

YOUR

SEPTEMBER 1988 £1.11

COMMODORE

BATMAN

Waiting for a dark knight

GREENPEACE

Fight for the foliage

GAMES

- Thunderbirds
- Rockstar ate my Hamster
- Escape from Drill
- Phobia



LICENCE TO KILL

0017 - On his own and out for revenge

ISSN 0261-8277



THE ULTIMATE BACKUP CARTRIDGE GOES ONE BETTER...
NEW ACTION REPLAY Mk V
 FOR CBM64/128 HAS ARRIVED

ONLY
£34.99
 POST FREE

AR V

by Commodore

EVEN MORE POWERFUL, MORE FRIENDLY & NOW EVEN MORE FEATURES!

ALL FEATURES AVAILABLE TO TAPE OR DISK

AR V is the most powerful backup system available for the Commodore 64/128. It's the only backup system that gives you the choice of backing up to tape or disk. It's the only backup system that gives you the choice of backing up to a single or double density disk. It's the only backup system that gives you the choice of backing up to a single or double density tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape.

AR V is the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape.

AR V is the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape.

AR V is the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape. It's the only backup system that gives you the choice of backing up to a single or double density disk or tape.

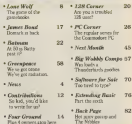
PERFORMANCE PROMISE

Commodore Display has an unrivalled range of industrial features. When you buy Action Replay V (2 year fixed) that you choose to be free. There's nothing it's reliable. It's easy to use. It's reliable.

GRAPHICS SUPPORT UTILITIES DISK

£12.99 Now your Commodore comes to a whole new type of display.

£12.99 A unique utility to allow you to take any part of your picture in 1/2 inch, 1/4 inch, 1/8 inch, 1/16 inch, 1/32 inch, 1/64 inch, 1/128 inch, 1/256 inch, 1/512 inch, 1/1024 inch, 1/2048 inch, 1/4096 inch, 1/8192 inch, 1/16384 inch, 1/32768 inch, 1/65536 inch, 1/131072 inch, 1/262144 inch, 1/524288 inch, 1/1048576 inch, 1/2097152 inch, 1/4194304 inch, 1/8388608 inch, 1/16777216 inch, 1/33554432 inch, 1/67108864 inch, 1/134217728 inch, 1/268435456 inch, 1/536870912 inch, 1/1073741824 inch, 1/2147483648 inch, 1/4294967296 inch, 1/8589934592 inch, 1/17179869184 inch, 1/34359738368 inch, 1/68719476736 inch, 1/137438953472 inch, 1/274877906944 inch, 1/549755813888 inch, 1/1099511627776 inch, 1/2199023255552 inch, 1/4398046511104 inch, 1/8796093022208 inch, 1/17592186044416 inch, 1/35184372088832 inch, 1/70368744177664 inch, 1/140737488355328 inch, 1/281474976710656 inch, 1/562949953421312 inch, 1/1125899906842624 inch, 1/2251799813685248 inch, 1/4503599627370496 inch, 1/9007199254740992 inch, 1/18014398509481984 inch, 1/36028797018963968 inch, 1/72057594037927936 inch, 1/144115188075855872 inch, 1/288230376151711744 inch, 1/576460752303423488 inch, 1/1152921504606846976 inch, 1/2305843009213693952 inch, 1/4611686018427387904 inch, 1/9223372036854775808 inch, 1/18446744073709551616 inch, 1/36893488147419103232 inch, 1/73786976294838206464 inch, 1/147573952589676412928 inch, 1/295147905179352825856 inch, 1/590295810358705651712 inch, 1/1180591620717411303424 inch, 1/2361183241434822606848 inch, 1/4722366482869645213696 inch, 1/9444732965739290427392 inch, 1/18889465931478580854784 inch, 1/37778931862957161709568 inch, 1/75557863725914323419136 inch, 1/151115727451828646838272 inch, 1/302231454903657293676544 inch, 1/604462909807314587353088 inch, 1/1208925819614629174706176 inch, 1/2417851639229258349412352 inch, 1/4835703278458516698824704 inch, 1/9671406556917033397649408 inch, 1/19342813113834066795298816 inch, 1/38685626227668133590597632 inch, 1/77371252455336267181195264 inch, 1/154742504910672534362390528 inch, 1/309485009821345068724781056 inch, 1/618970019642690137449562112 inch, 1/1237940039285380274899124224 inch, 1/2475880078570760549798248448 inch, 1/4951760157141521099596496896 inch, 1/9903520314283042199192993792 inch, 1/19807040628566084398385987584 inch, 1/39614081257132168796771975168 inch, 1/79228162514264337593543950336 inch, 1/158456325028528675187087900672 inch, 1/316912650057057350374175801344 inch, 1/633825300114114700748351602688 inch, 1/1267650600228229401496703205376 inch, 1/2535301200456458802993406410752 inch, 1/5070602400912917605986812821504 inch, 1/10141204801825835211973625643008 inch, 1/20282409603651670423947251286016 inch, 1/40564819207303340847894502572032 inch, 1/81129638414606681695789005144064 inch, 1/162259276829213363391578010288128 inch, 1/324518553658426726783156020576256 inch, 1/649037107316853453566312041152512 inch, 1/1298074214633706907132624082305024 inch, 1/2596148429267413814265248164610048 inch, 1/5192296858534827628530496329220096 inch, 1/10384593717069655257060992658440192 inch, 1/20769187434139310514121985316880384 inch, 1/41538374868278621028243970633760768 inch, 1/83076749736557242056487941267521536 inch, 1/166153499473114484112975882535043072 inch, 1/332306998946228968225951765070086144 inch, 1/664613997892457936451903530140172288 inch, 1/1329227995784915872903807060280344576 inch, 1/2658455991569831745807614120560689152 inch, 1/5316911983139663491615228241121378304 inch, 1/10633823966279326983230456482242756608 inch, 1/21267647932558653966460912964485513216 inch, 1/42535295865117307932921825928971026432 inch, 1/85070591730234615865843651857942052864 inch, 1/170141183460469231731687303715884105728 inch, 1/340282366920938463463374607431768211456 inch, 1/680564733841876926926749214863536422912 inch, 1/1361129467683753853853498429727072845824 inch, 1/2722258935367507707706996859454145691648 inch, 1/5444517870735015415413993718908291383296 inch, 1/10889035741470030830827987437816582766592 inch, 1/21778071482940061661655974875633165533184 inch, 1/43556142965880123323311949751266331066368 inch, 1/87112285931760246646623899502532662132736 inch, 1/174224571863520493293247799005065244265472 inch, 1/348449143727040986586495598010130488530944 inch, 1/696898287454081973172991196020260977061888 inch, 1/1393796574908163946345982392040521954123776 inch, 1/2787593149816327892691964784081043908247552 inch, 1/5575186299632655785383929568162087816495104 inch, 1/11150372599265311570767859136324175632990208 inch, 1/22300745198530623141535718272648351265980416 inch, 1/44601490397061246283071436545296702531960832 inch, 1/89202980794122492566142873090593405063921664 inch, 1/178405961588244985132285746181186810127843328 inch, 1/356811923176489970264571492362373620255686656 inch, 1/713623846352979940529142984724747240511373312 inch, 1/1427247692705959881058285969449494481022746624 inch, 1/2854495385411919762116571938898988962045493248 inch, 1/5708990770823839524233143877797977924090986496 inch, 1/11417981541647679048466287755595955848181972992 inch, 1/22835963083295358096932575511191911696363945984 inch, 1/45671926166590716193865151022383823392727891968 inch, 1/91343852333181432387730302044767646785455783936 inch, 1/182687704666362864775460604089535293570911567872 inch, 1/365375409332725729550921208179070587141823135744 inch, 1/730750818665451459101842416358141174283646271488 inch, 1/1461501637330902918203684832716282348567292542976 inch, 1/2923003274661805836407369665432564697134585085952 inch, 1/5846006549323611672814739330865129394269170171904 inch, 1/11692013098647223345629478661730258788538340343808 inch, 1/23384026197294446691258957323460517577076680687616 inch, 1/46768052394588893382517914646921035154153361375232 inch, 1/93536104789177786765035829293842070308306722750464 inch, 1/187072209578355573530071658587684140616613445500928 inch, 1/374144419156711147060143317175368281233226891001856 inch, 1/748288838313422294120286634350736562466453782003712 inch, 1/1496577676626844588240573268701473124932907564007424 inch, 1/2993155353253689176481146537402946249865815128014848 inch, 1/5986310706507378352962293074805892499731630256029696 inch, 1/11972621413014756705924586149611784999463260512059392 inch, 1/23945242826029513411849172299223569998926521024118784 inch, 1/47890485652059026823698344598447139997853042048237568 inch, 1/95780971304118053647396689196894279995706084096475136 inch, 1/191561942608236107294793378393788559991412168192950272 inch, 1/383123885216472214589586756787577119982824336385900544 inch, 1/766247770432944429179173513575154239965648672771801088 inch, 1/1532495540865888858358347027150308479931297345543602176 inch, 1/3064991081731777716716694054300616959862594691087204352 inch, 1/6129982163463555433433388108601233919725189382174408704 inch, 1/1225996432692711086686677621720246783945037876434881738 inch, 1/2451992865385422173373355243440493567890075752869763472 inch, 1/4903985730770844346746710486880987135780151505739526944 inch, 1/9807971461541688693493420973761974271560303011479053888 inch, 1/19615942923083377386986841947523948543120606022958107776 inch, 1/39231885846166754773973683895047897086241212045916215552 inch, 1/78463771692333509547947367790095794172482424091832431104 inch, 1/156927543384667019095894735580191588344964848183664862208 inch, 1/313855086769334038191789471160383176689929696367329724416 inch, 1/627710173538668076383578942320766353379859392734659448832 inch, 1/1255420347077336152767157884641532706759718785469318897664 inch, 1/2510840694154672305534315769283065413519437570938637795328 inch, 1/5021681388309344611068631538566130827038875141877275590656 inch, 1/10043362776618689222137263077132261654077750283754551181312 inch, 1/20086725553237378444274526154264523308155500567509102362624 inch, 1/40173451106474756888549052308529046616311001135018204725248 inch, 1/80346902212949513777098104617058093232622002270036409450496 inch, 1/160693804425899027554196209234116186465244004540072818900992 inch, 1/321387608851798055108392418468232372930488009080145637801984 inch, 1/642775217703596110216784836936464745860976018160291275603968 inch, 1/1285550435407192220433569673872929491721952036320582551207936 inch, 1/2571100870814384440867139347745858983443904072641165102415872 inch, 1/5142201741628768881734278695491717966887808145282330204831744 inch, 1/10284403483257537763468557390983435933775616290564660409663488 inch, 1/20568806966515075526937114781966871867551232581129320819326976 inch, 1/41137613933030151053874229563933743735102465162258641638653952 inch, 1/82275227866060302107748459127867487470204930324517283277307904 inch, 1/164550455732120604215496918255734974940409860649034566554615808 inch, 1/329100911464241208430993836511469949880819721298069133109231616 inch, 1/658201822928482416861987673022939899761639442596138266218463232 inch, 1/1316403645856964833723975346045879799523278885192276532436926464 inch, 1/2632807291713929667447950692091759599046557770384553064873852928 inch, 1/5265614583427859334895901384183519198093115540769106129747705856 inch, 1/10531229166855718669791802768367038396186231081538212259495411712 inch, 1/21062458333711437339583605536734076792372462163076424518990823424 inch, 1/42124916667422874679167211073468153584744924326152849037981646848 inch, 1/8424983333484574935833442214693630716948984865230569807596329376 inch, 1/16849966668969149871666884429387261433897969730461139615192658752 inch, 1/33699933337938299743333768858774522867795939460922279230385317504 inch, 1/67399866675876599486667537717549045735591878921844558460770635008 inch, 1/134799733351753198973335075435098091471183757843689116921541270016 inch, 1/269599466703506397946670150870196182942367515687378233843082540032 inch, 1/539198933407012795893340301740392365884735031374756467686165080064 inch, 1/1078397866814025591786680603480784731769470062749512935372330160128 inch, 1/2156795733628051183573361206961569463538940125499025870744660320256 inch, 1/4313591467256102367146722413923138927077880250998051741489320640512 inch, 1/8627182934512204734293444827846277854155760501996103482978641281024 inch, 1/17254365869024409468586889655692555708311521003992206965973282562048 inch, 1/34508731738048818937173779311385111416623042007984413931946565124096 inch, 1/69017463476097637874347558622770222833246084015968827863893130248192 inch, 1/138034926952195275748695117245540445666492168031937655727786260496384 inch, 1/276069853904390551497390234491080891332984336063875311455572520992768 inch, 1/552139707808781102994780468982161782665968672127750622911145041985536 inch, 1/1104279415617562205989560937964323565331937344255501245822290083971072 inch, 1/2208558831235124411979121875928647130663874688511002491644580167942144 inch, 1/4417117662470248823958243751857294261327749377022004983289160335884288 inch, 1/8834235324940497647916487503714588522655498754044009966578320671768576 inch, 1/17668470649880995295832975007429177045310997508088019933156641343537152 inch, 1/35336941299761990591665950014858354090621995016176039866313282687074304 inch, 1/70673882599523981183331900029716708181243990032352079732626565374148608 inch, 1/141347765199047962366663800059433416362487980064704159465253130748297216 inch, 1/282695530398095924733327600118866832724975960129408318930506261496594432 inch, 1/565391060796191849466655200237733665449951920258816637861012522993188864 inch, 1/1130782121592383698933310400475467330899903840517633275722025045986377728 inch, 1/2261564243184767397866620800950934661799807681035266551444050091972755456 inch, 1/4523128486369534795733241601901869323599615362070533102888100183945510912 inch, 1/9046256972739069591466483203803738647199230724141066205776200367891021824 inch, 1/18092513945478139182932966407607477294398461448282132411552400735782043648 inch, 1/36185027890956278365865932815214954588796922896564264823104801471564087296 inch, 1/72370055781912556731731865630429909177593845793128529646209602943128174592 inch, 1/144740111563825113463463731260859818355187691586257059292419205886256349184 inch, 1/289480223127650226926927462521719636710375383172514118584838411772512698368 inch, 1/578960446255300453853854925043439273420750766345028237169676823545025396736 inch, 1/1157920892510600907707709850086878546841501532690056474339353647090050793472 inch, 1/231584178502120181541541

[illegible]

<p>• Chaps 29 Get your hands round your tail!</p> <p>• Means and Squares 42 Tutorial into mathematical formulae</p> <p>• Scrambling on the F28 46 Continuation of the expert system</p>	<p>• Prime FX 50 Correction for our previous errors</p> <p>• POS B.I 54 Enhanced POS commands for disk users</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------

Editorial

If you flip through the pages of this month's *Four Commodore* you may find a few subtle (and a couple of blatantly obvious) changes. First is because the editorial team upon my arrival to the fold, has decided to offer a brighter future for PC: one that we can all enjoy together.

My first major decision was to render that scene around Burnside, and I hope you all agree that that was a wise move. Apparently the movie grossed over \$40,000,000 in six first weekend, and ending it at just \$10m less to make it less sure that depleted any fans from Warner Brothers that it was to flop, and that is before it reaches moviegoers. Further on at that time you can read about the reasons the film was underwritten, and a review of the movie.

Other features of note are an *Lecture to Kill* - the new Bond film, *Graveyard* and *Lost Wolf*, so make sure you're got your robot at play on. They are **FREE!**

Although there are many commentators based online we haven't left out the usual suspects, or the machine-wired columns. The programs can now be found in a section of with their own, so you can keep them separate from the rest.

I hope you enjoy this meal as much as we're enjoying putting it together and that leaves me to say welcome to the Peninsula.

Age Group	Total	Male	Female	Unknown
18-24	12%	10%	14%	10%
25-34	22%	20%	24%	20%
35-44	28%	26%	30%	26%
45-54	20%	18%	22%	18%
55-64	15%	14%	16%	14%
65+	5%	4%	6%	4%

Deputy Editor: Richard
Ben Mendenhall

**Technical Editor and
Builder: Jim**

Strong Edition. Heavy
Durable Cover.

Advertisement Manager,
David M. Broun, Jr., President

Source: *U.S. Census Bureau, Current Population Reports, 1990*

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 111–116

Figure 6

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

Link: <http://www.ck12.org/Book-Search/Book-Search.aspx?c=1&q=biology&v=1>

1. *Journal of the American Medical Association*, 1997; 277: 1039-1043.

100

Investigating Student Learning

Abstract

Background

Translation: Chinese: 1992

The contents of this publication, including all articles, designs, drawings and programs and all copyright and other intellectual property rights therein, belong to Argus Specialized Publications Limited. All rights reserved by the Law of Copyright and other and broad property rights and to extent of intellectual property laws are specifically reserved to Argus Specialized Publications Limited and any reproduction

Data Statements

The Bug Bites Back

Now that The Rug, the modestly successful computer dance has officially folded, artists James Lewis and Jeffrey Derry are to sell their entire software review library to raise funds for charity.

The collection includes 400 years of software (games and otherwise) and a majority of the profits are to go to a consortium of spacious charities and voluntary organizations.

If you wish for more information, the current list, etc. write to 28 Leaside Avenue, Murrell Hill, London NW3 1BU. Please include an SAE.

EA's Horror Score

Electronic Arts is set to bring the horrific world of *H. H. Lovecraft's Cthulhu* mythos onto Amiga and PC screens in September. The *Heard of the Shogun* is a role playing game set in the coming 20c, in which the player must use intelligence and cunning in order to unravel the secrets of ancient lore and discover many secrets before falling beneath seemingly normal life.

The game system that was developed by board game designers and former western, Edwih Gurnee, allows the player to create a character by selecting one, ultimately one of six professions, and proficiency in over 20 skills. There will directly affect the outcome and his success in the game.

The *House of the Shadow* will appear first on the PC in early September and later on the Amiga. Both versions will cost £24.99.



requester and power window operation
and door latching. 1993.

Quadrant EM Technology is
Lightem Corp. Road, Laguna
Serra, CA. Product by Citrus
High Performance.

Expenditures categorized as overhead are the expenses of the government and not necessarily those of the program. What every effort is made in determining actual program expenditures the source we attempt to hold responsible for any errors, that is, the program.

Your Computer, every morning
 Your 64 is a totally responsive
 Response, no other kind of Response
 with words. Answer Questions
 Right now. (Call 1-800-333-3333)

Advertising Sales Manager
 Advertising Sales Office, 100
 Community College Blvd.
 University City, San Diego 92161
 (619) 594-1111 ext. 222
 Fax: (619) 594-1111 ext. 222
 E-mail: advertising@ucsd.edu
 Web: www.ucsd.edu

OFFICIAL COMMODORE RANGA (S) 484

Computer	Price
Amiga 1000	£299.00
Amiga 500	£249.00
Amiga 2000	£499.00
Amiga 500+ (with 1MB memory) - £299.00	£299.00
Amiga 500+ (with 2MB memory) - £349.00	£349.00
Amiga 500+ (with 4MB memory) - £399.00	£399.00
Amiga 500+ (with 8MB memory) - £449.00	£449.00
Amiga 500+ (with 16MB memory) - £499.00	£499.00
Amiga 500+ (with 32MB memory) - £549.00	£549.00
Amiga 500+ (with 64MB memory) - £599.00	£599.00
Amiga 500+ (with 128MB memory) - £649.00	£649.00
Amiga 500+ (with 256MB memory) - £699.00	£699.00
Amiga 500+ (with 512MB memory) - £749.00	£749.00
Amiga 500+ (with 1024MB memory) - £799.00	£799.00
Amiga 500+ (with 2048MB memory) - £849.00	£849.00
Amiga 500+ (with 4096MB memory) - £899.00	£899.00
Amiga 500+ (with 8192MB memory) - £949.00	£949.00
Amiga 500+ (with 16384MB memory) - £999.00	£999.00
Amiga 500+ (with 32768MB memory) - £1049.00	£1049.00
Amiga 500+ (with 65536MB memory) - £1099.00	£1099.00
Amiga 500+ (with 131072MB memory) - £1149.00	£1149.00
Amiga 500+ (with 262144MB memory) - £1199.00	£1199.00
Amiga 500+ (with 524288MB memory) - £1249.00	£1249.00
Amiga 500+ (with 1048576MB memory) - £1299.00	£1299.00
Amiga 500+ (with 2097152MB memory) - £1349.00	£1349.00
Amiga 500+ (with 4194304MB memory) - £1399.00	£1399.00
Amiga 500+ (with 8388608MB memory) - £1449.00	£1449.00
Amiga 500+ (with 16777216MB memory) - £1499.00	£1499.00
Amiga 500+ (with 33554432MB memory) - £1549.00	£1549.00
Amiga 500+ (with 67108864MB memory) - £1599.00	£1599.00
Amiga 500+ (with 134217728MB memory) - £1649.00	£1649.00
Amiga 500+ (with 268435456MB memory) - £1699.00	£1699.00
Amiga 500+ (with 536870912MB memory) - £1749.00	£1749.00
Amiga 500+ (with 1073741824MB memory) - £1799.00	£1799.00
Amiga 500+ (with 2147483648MB memory) - £1849.00	£1849.00
Amiga 500+ (with 4294967296MB memory) - £1899.00	£1899.00
Amiga 500+ (with 8589934592MB memory) - £1949.00	£1949.00
Amiga 500+ (with 17179869184MB memory) - £1999.00	£1999.00
Amiga 500+ (with 34359738368MB memory) - £2049.00	£2049.00
Amiga 500+ (with 68719476736MB memory) - £2099.00	£2099.00
Amiga 500+ (with 137438953472MB memory) - £2149.00	£2149.00
Amiga 500+ (with 274877906944MB memory) - £2199.00	£2199.00
Amiga 500+ (with 549755813888MB memory) - £2249.00	£2249.00
Amiga 500+ (with 1099511627776MB memory) - £2299.00	£2299.00
Amiga 500+ (with 2199023255552MB memory) - £2349.00	£2349.00
Amiga 500+ (with 4398046511104MB memory) - £2399.00	£2399.00
Amiga 500+ (with 8796093022208MB memory) - £2449.00	£2449.00
Amiga 500+ (with 17592186044416MB memory) - £2499.00	£2499.00
Amiga 500+ (with 35184372088832MB memory) - £2549.00	£2549.00
Amiga 500+ (with 70368744177664MB memory) - £2599.00	£2599.00
Amiga 500+ (with 140737488355328MB memory) - £2649.00	£2649.00
Amiga 500+ (with 281474976710656MB memory) - £2699.00	£2699.00
Amiga 500+ (with 562949953421312MB memory) - £2749.00	£2749.00
Amiga 500+ (with 1125899906842624MB memory) - £2799.00	£2799.00
Amiga 500+ (with 2251799813685248MB memory) - £2849.00	£2849.00
Amiga 500+ (with 4503599627370496MB memory) - £2899.00	£2899.00
Amiga 500+ (with 9007199254740992MB memory) - £2949.00	£2949.00
Amiga 500+ (with 18014398509481984MB memory) - £2999.00	£2999.00
Amiga 500+ (with 36028797018963968MB memory) - £3049.00	£3049.00
Amiga 500+ (with 72057594037927936MB memory) - £3099.00	£3099.00
Amiga 500+ (with 144115188075855872MB memory) - £3149.00	£3149.00
Amiga 500+ (with 288230376151711744MB memory) - £3199.00	£3199.00
Amiga 500+ (with 576460752303423488MB memory) - £3249.00	£3249.00
Amiga 500+ (with 1152921504606846976MB memory) - £3299.00	£3299.00
Amiga 500+ (with 2305843009213693952MB memory) - £3349.00	£3349.00
Amiga 500+ (with 4611686018427387904MB memory) - £3399.00	£3399.00
Amiga 500+ (with 9223372036854775808MB memory) - £3449.00	£3449.00
Amiga 500+ (with 18446744073709551616MB memory) - £3499.00	£3499.00
Amiga 500+ (with 36893488147419103232MB memory) - £3549.00	£3549.00
Amiga 500+ (with 73786976294838206464MB memory) - £3599.00	£3599.00
Amiga 500+ (with 147573952589676412928MB memory) - £3649.00	£3649.00
Amiga 500+ (with 295147905179352825856MB memory) - £3699.00	£3699.00
Amiga 500+ (with 590295810358705651712MB memory) - £3749.00	£3749.00
Amiga 500+ (with 1180591620717411303424MB memory) - £3799.00	£3799.00
Amiga 500+ (with 2361183241434822606848MB memory) - £3849.00	£3849.00
Amiga 500+ (with 4722366482869645213696MB memory) - £3899.00	£3899.00
Amiga 500+ (with 9444732965739290427392MB memory) - £3949.00	£3949.00
Amiga 500+ (with 18889465931478580854784MB memory) - £3999.00	£3999.00
Amiga 500+ (with 37778931862957161709568MB memory) - £4049.00	£4049.00
Amiga 500+ (with 75557863725914323419136MB memory) - £4099.00	£4099.00
Amiga 500+ (with 151115727451828646838272MB memory) - £4149.00	£4149.00
Amiga 500+ (with 302231454903657293676544MB memory) - £4199.00	£4199.00
Amiga 500+ (with 604462909807314587353088MB memory) - £4249.00	£4249.00
Amiga 500+ (with 1208925819614629174706176MB memory) - £4299.00	£4299.00
Amiga 500+ (with 2417851639229258349412352MB memory) - £4349.00	£4349.00
Amiga 500+ (with 4835703278458516698824704MB memory) - £4399.00	£4399.00
Amiga 500+ (with 9671406556917033397649408MB memory) - £4449.00	£4449.00
Amiga 500+ (with 19342813113834066795298816MB memory) - £4499.00	£4499.00
Amiga 500+ (with 38685626227668133590597632MB memory) - £4549.00	£4549.00
Amiga 500+ (with 77371252455336267181195264MB memory) - £4599.00	£4599.00
Amiga 500+ (with 154742504910672534362390528MB memory) - £4649.00	£4649.00
Amiga 500+ (with 309485009821345068724781056MB memory) - £4699.00	£4699.00
Amiga 500+ (with 618970019642690137449562112MB memory) - £4749.00	£4749.00
Amiga 500+ (with 1237940039285380274899124224MB memory) - £4799.00	£4799.00
Amiga 500+ (with 2475880078570760549798248448MB memory) - £4849.00	£4849.00
Amiga 500+ (with 4951760157141521099596496896MB memory) - £4899.00	£4899.00
Amiga 500+ (with 9903520314283042199192993792MB memory) - £4949.00	£4949.00
Amiga 500+ (with 19807040628566084398385987584MB memory) - £4999.00	£4999.00
Amiga 500+ (with 39614081257132168796771975168MB memory) - £5049.00	£5049.00
Amiga 500+ (with 79228162514264337593543950336MB memory) - £5099.00	£5099.00
Amiga 500+ (with 158456325028528675187087900672MB memory) - £5149.00	£5149.00
Amiga 500+ (with 316912650057057350374175801344MB memory) - £5199.00	£5199.00
Amiga 500+ (with 633825300114114700748351602688MB memory) - £5249.00	£5249.00
Amiga 500+ (with 1267650600228229401496703205376MB memory) - £5299.00	£5299.00
Amiga 500+ (with 2535301200456458802993406410752MB memory) - £5349.00	£5349.00
Amiga 500+ (with 5070602400912917605986812821504MB memory) - £5399.00	£5399.00
Amiga 500+ (with 10141204801825835211973625643008MB memory) - £5449.00	£5449.00
Amiga 500+ (with 20282409603651670423947251286016MB memory) - £5499.00	£5499.00
Amiga 500+ (with 40564819207303340847894502572032MB memory) - £5549.00	£5549.00
Amiga 500+ (with 81129638414606681695789005144064MB memory) - £5599.00	£5599.00
Amiga 500+ (with 162259276829213363391578010288128MB memory) - £5649.00	£5649.00
Amiga 500+ (with 324518553658426726783156020576256MB memory) - £5699.00	£5699.00
Amiga 500+ (with 649037107316853453566312041152512MB memory) - £5749.00	£5749.00
Amiga 500+ (with 1298074214633706907132624082305024MB memory) - £5799.00	£5799.00
Amiga 500+ (with 2596148429267413814265248164610048MB memory) - £5849.00	£5849.00
Amiga 500+ (with 5192296858534827628530496329220096MB memory) - £5899.00	£5899.00
Amiga 500+ (with 10384593717069655257060992658440192MB memory) - £5949.00	£5949.00
Amiga 500+ (with 20769187434139310514121985316880384MB memory) - £5999.00	£5999.00
Amiga 500+ (with 41538374868278621028243970633760768MB memory) - £6049.00	£6049.00
Amiga 500+ (with 83076749736557242056487941267521536MB memory) - £6099.00	£6099.00
Amiga 500+ (with 166153499473114484112975882535043072MB memory) - £6149.00	£6149.00
Amiga 500+ (with 332306998946228968225951765070086144MB memory) - £6199.00	£6199.00
Amiga 500+ (with 664613997892457936451903530140172288MB memory) - £6249.00	£6249.00
Amiga 500+ (with 1329227995784915872903807060280344576MB memory) - £6299.00	£6299.00
Amiga 500+ (with 2658455991569831745807614120560689152MB memory) - £6349.00	£6349.00
Amiga 500+ (with 5316911983139663491615228241121378304MB memory) - £6399.00	£6399.00
Amiga 500+ (with 10633823966279326983230456482242756608MB memory) - £6449.00	£6449.00
Amiga 500+ (with 21267647932558653966460912964485513216MB memory) - £6499.00	£6499.00
Amiga 500+ (with 42535295865117307932921825928971026432MB memory) - £6549.00	£6549.00
Amiga 500+ (with 85070591730234615865843651857942052864MB memory) - £6599.00	£6599.00
Amiga 500+ (with 170141183460469231731687303715884105728MB memory) - £6649.00	£6649.00
Amiga 500+ (with 340282366920938463463374607431768211456MB memory) - £6699.00	£6699.00
Amiga 500+ (with 680564733841876926926749214863536422912MB memory) - £6749.00	£6749.00
Amiga 500+ (with 1361129467683753853853498429727072845824MB memory) - £6799.00	£6799.00
Amiga 500+ (with 272225893536750770770699685945414569152MB memory) - £6849.00	£6849.00
Amiga 500+ (with 544451787073501541541399371890829138304MB memory) - £6899.00	£6899.00
Amiga 500+ (with 1088903574147003083082798743781658276608MB memory) - £6949.00	£6949.00
Amiga 500+ (with 2177807148294006166165597487563316553216MB memory) - £6999.00	£6999.00
Amiga 500+ (with 4355614296588012332331194975126633106432MB memory) - £7049.00	£7049.00
Amiga 500+ (with 8711228593176024664662389950253266212864MB memory) - £7099.00	£7099.00
Amiga 500+ (with 17422457186352049329324779900506532425728MB memory) - £7149.00	£7149.00
Amiga 500+ (with 34844914372704098658649559801013064851456MB memory) - £7199.00	£7199.00
Amiga 500+ (with 69689828745408197317299119602026129702912MB memory) - £7249.00	£7249.00
Amiga 500+ (with 139379657490816394634598239204052259405824MB memory) - £7299.00	£7299.00
Amiga 500+ (with 278759314981632789269196478408104518811648MB memory) - £7349.00	£7349.00
Amiga 500+ (with 557518629963265578538392956816209037623296MB memory) - £7399.00	£7399.00
Amiga 500+ (with 1115037259926531157076785913632418075246592MB memory) - £7449.00	£7449.00
Amiga 500+ (with 2230074519853062314153571827264836150493184MB memory) - £7499.00	£7499.00
Amiga 500+ (with 446014903970612462830714365452967230098656MB memory) - £7549.00	£7549.00
Amiga 500+ (with 892029807941224925661428730905934460197312MB memory) - £7599.00	£7599.00
Amiga 500+ (with 1784059615882449851322857461811868920394624MB memory) - £7649.00	£7649.00
Amiga 500+ (with 3568119231764899702645714923623737840789248MB memory) - £7699.00	£7699.00
Amiga 500+ (with 7136238463529799405291429847247475681578496MB memory) - £7749.00	£7749.00
Amiga 500+ (with 14272476927059598810582859694494951363156992MB memory) - £7799.00	£7799.00
Amiga 500+ (with 28544953854119197621165719388989902726313984MB memory) - £7849.00	£7849.00
Amiga 500+ (with 57089907708238395242331438777979805452627968MB memory) - £7899.00	£7899.00
Amiga 500+ (with 114179815416476790484662877555959610905255936MB memory) - £7949.00	£7949.00
Amiga 500+ (with 228359630832953580969325755111919221810511872MB memory) - £7999.00	£7999.00
Amiga 500+ (with 456719261665907161938651510223838443621023744MB memory) - £8049.00	£8049.00
Amiga 500+ (with 913438523331814323877303020447676887242047488MB memory) - £8099.00	£8099.00
Amiga 500+ (with 1826877046663628647754606040895353774484094976MB memory) - £8149.00	£8149.00
Amiga 500+ (with 3653754093327257295509212081790707548968189952MB memory) - £8199.00	£8199.00
Amiga 500+ (with 7307508186654514591018424163581415097936379904MB memory) - £8249.00	£8249.00
Amiga 500+ (with 14615016373309029182036848327162830195872759808MB memory) - £8299.00	£8299.00
Amiga 500+ (with 29230032746618058364073696654325660391745519616MB memory) - £8349.00	£8349.00
Amiga 500+ (with 58460065493236116728147393308651320783491039232MB memory) - £8399.00	£8399.00
Amiga 500+ (with 116920130986472233456294786617302641566982078464MB memory) - £8449.00	£8449.00
Amiga 500+ (with 233840261972944466912589573234605283133964156928MB memory) - £8499.00	£8499.00
Amiga 500+ (with 467680523945888933825179146469210566267928313856MB memory) - £8549.00	£8549.00
Amiga 500+ (with 935361047891777867650358292938421132535856627712MB memory) - £8599.00	£8599.00
Amiga 500+ (with 1870722095783555735300716585876842265071713255424MB memory) - £8649.00	£8649.00
Amiga 500+ (with 3741444191567111470601433171753684530143426510848MB memory) - £8699.00	£8699.00
Amiga 500+ (with 7482888383134222941202866343507369060286853021696MB memory) - £8749.00	£8749.00
Amiga 500+ (with 1496577676626844588240573268701473812057370	

Trendy Letties in Fun School Controversy

Fun School 2, Database's excellent tracking software has been rejected for use in schools by 'Trendy' educationists, who believe that it emphasises the 3Rs too much.

The 3Rs (Reading, Writing, and 'Arithmetic') are currently being worked out of the national curriculum by such left-wing educationists as the 'old Labour' methods of teaching and

Peter Dawkins, one of the co-authors of the software states "What they don't like about it - apart from it being based on the 3Rs - is that it involves an element of competition, with a tick marking up of the questions answered correctly".

Shirley Osborn, a teacher who has been using Fun School 2 in her class at Paplar Street Primary School, Andoverham, has written to the Minister of Education Kenneth Baker, to urge him to explore alternative

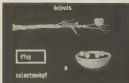
PC Formulae One.

Electronic Arts' Formula One PC version will see the racing *Amiga* game spin-off *Formula Formula* first. Planned for an early Spring date, the game will bring together the excitement of driving in a Formula One competition against Nigel Mansell, Nigel Mansell and Nelson Piquet, and the challenge of managing a team.

The game will feature all six tracks from the 1988 season including Monaco, Silverstone, and Suzuka.

Formula Formula features both the current season's cars and the current season's drivers in a simulated Formula One racing game. The game will feature all six tracks from the 1988 season including Monaco, Silverstone, and Suzuka.

The game will feature all six tracks from the 1988 season including Monaco, Silverstone, and Suzuka.



Socket to Them

Computers, monitors and TVs are all vulnerable to mains interference. This can damage hardware, corrupt data, and cause games to crash, not ideal when the high score is about to be broken. The most common type of interference is a mains 'spike', provoked by home equipment such as washing machines, power tools, and microwave ovens. The same applies with your neighbour's electrical equipment, or even the electricity company having problems maintaining a stable voltage.

One solution to such a problem comes from Apollo Electronics Products Ltd who have introduced a new Apollo Spike Suppressor which, for the price of £12.99



for a plug unit, can protect your computer system. These British made devices are simple to use, the most you have to do is replace the main plug.

The range also includes a 3-way adaptor for £16 and a 4-gang strip for £35 (prices include P&P and VAT). All the units are capable of absorbing a spike of 4000 Amps/25 Joules.

For more information contact Apollo Electronics Products Ltd on (0483) 8156.

Guns, Set and Mix

Your Commodore are joined to prevent mouse theft and dust covers for the A500 and C128. The covers are especially designed to protect your computer from dust, grime, inlets, via, Hb, ensuring your console do not get clogged by spatters and internal acid rain.

Made exclusively for your favourite computer mix, they are stylish and robust, and come complete with

their own only logo.

The mouse cover has are emblazoned with the 'Your Commodore' logo, and they help to allow your mouse grip on any surface.

At the price of £4.45 for the covers and £4.45 for the Mouse Mix, they represent terrific value for money.

They are available from: **Readies, Barlows, Angus House, Boundary Way, Hemel Hempstead HP1 7BT**



It's A-Maze-ing

Hewlett's latest game sounds our way to late August. Titled Macromaze and featuring a character called Flippo who must travel through the maze of Tangle Field (name, trip, so he goes) - hence the title Flippo.

Adding alone, heider has spent, but there are some to supply him with extra power. Once granted, the maze will lock away to him. Flippo will see another in

action.

The Amiga version will have 25 levels to complete, while the C64 will only have 12-15. The price of the C64 version will be (as usual) \$9.99.



Anyone for Tennis

Watershed, under their Imageworks label, has secured the rights for the Japanese hit, Puring Shot. Programmed by Tropic, the series behind Thunderbolt and Mastermind, among others, it captures all the aspects of the arcade

machine and will be available on both the C64 and the Amiga.

Puring Shot features doubles and singles championships, taking place on clay and grass courts, in an effort to win the elusive Grand Slam. It offers an impressive variety of game play options with the player controlling both the pos-

itioning, and type of shot taken. (oh, you see?) The computer is available to follow the action and then act as a perspective to measure if keep the game overhead for complete 3D for some.

Bill Burt, the net judge and the lecturer are all there and you can expect to see it in late August.

Oil See You in the Morning

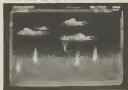
Rainbow Labs, the company responsible for the wonderful one Demos/Karaoke, has undertaken a different kind of project.

Released on the Rainbow Software label, Oil Impersonation deals with the oil and threat world of the oil business. You can become JR

as you try to put your competitors out of business. When it even better is that these enemies can be your friends, or the computer.

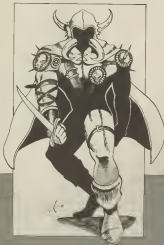
Bullylog, drilling, trading, upgrading and managing are all involved in getting to the top, so only the devious minds may apply.

Oil Impersonation will be available in September on the C64 and Amiga.



Only the Lonely

The Lone Wolf handwagon continues to roll along the road of success. Rick Henderson hitches a ride and prepares for a one-way trip to fantasy land.



plus the downfall of the Ku, the warrior chief of Sontarland (as is quoted on the back). Although it contains many good fantasy ideas, there is always a feeling of déjà-vu connected.

The Dark Door Opens is not much better in theory, but perhaps a more enjoyable read. Now that the Ku have been eliminated, Lone Wolf is born and the plot follows his quest to reach King Ulmer and warn him of impending doom.

Anthogames Software has now raised the effect of Lone Wolf, and thus released a new computer game based on his exploits.

Entitled simply *The Mirror of Death*, the game places Lone Wolf in an arcade adventure. Although one could be forgiven for thinking that it has more than a smidgen of hack-and-slash action more usual in a best-seller.

The plot follows Lone Wolf as he tries to recover a stolen gem, once belonging to his mentors, the Ku. This is no easy task as the tower in which it is imprisoned is full of fire-spitting gnomes, dragons and wandering other fantasy creatures. It is time to turn

back to the drawing board. The game is a fairly simple affair, with a lot of cut-scenes and a lot of text. The graphics are simple, but the sound effects are good. The game is a good example of a computer game that is based on a book, and it is a good example of a computer game that is based on a book.

The game is a good example of a computer game that is based on a book, and it is a good example of a computer game that is based on a book. The game is a good example of a computer game that is based on a book, and it is a good example of a computer game that is based on a book.

The game is a good example of a computer game that is based on a book, and it is a good example of a computer game that is based on a book. The game is a good example of a computer game that is based on a book, and it is a good example of a computer game that is based on a book.

The new *Novus*, the way in which we write with it, is the way in which we write with it. *Novus* is the way in which we write with it.

1986

Last, but certainly not least, is the new Lone Wolf releases in the inter-



active telephone line, thanks to Joe Dever. This is surely in direct competition with Steve Jackson's Fax.

The company involved, *Broadsword*, has announced a new label under which they hope to release many games of this kind. *Forever of Ravens* being just the first. *Phoenician* (TM) games will use the same system of operation, although at this stage it is hard to determine whether they'll just be working with Lone Wolf, or if they may branch into other areas of miniature games.

Unlike Fax, which uses a new method to determine when *PGS* breaks are ground with voice-interactive technology. This allows the player to make 'real-time' decisions during play, and increases the tension created by the changing sound effects.

The way in which the music is used is also an added feature. The music is used to create a sense of tension and to create a sense of danger. The music is used to create a sense of danger and to create a sense of danger.

The music is used to create a sense of danger and to create a sense of danger. The music is used to create a sense of danger and to create a sense of danger.

The music is used to create a sense of danger and to create a sense of danger. The music is used to create a sense of danger and to create a sense of danger.

The music is used to create a sense of danger and to create a sense of danger. The music is used to create a sense of danger and to create a sense of danger.

The music is used to create a sense of danger and to create a sense of danger. The music is used to create a sense of danger and to create a sense of danger.

this is the price which must be charged. Off peak, rate is 25p per minute, whilst peak is 35p per minute, making a maximum £15 for a hour's play (off peak), which will add up to £100 per week, if it is played for an hour a day, quite a sum of money.

This is most worrying for the younger children who become addicted to such things (hence the reason for *Children's domain*). My advice to anybody worried about the cost, buy one of these gamebooks, at around £2.50, and get a friend around to read it out to you, it's not that much different and certainly a lot cheaper.

I'm not here to condemn the line, I think it's rather good idea, and if played in moderation it's a very worthwhile venture.

All the Lone Wolf products have something going for them, and the fact is that they'll be popular for a long while yet, thanks to Joe Dever's persistence, and *Broadsword* money for when both are flowing legends can be born.

Lone Wolf Bibliography

Lone Wolf Gamebooks

- 1 *Flight From The Dark* - Joe Dever & Gary Chalk
- 2 *Fire on the Water* - Joe Dever & Gary Chalk
- 3 *The Caverns of Ratin* - Joe Dever & Gary Chalk
- 4 *The Charn of Deven* - Joe Dever & Gary Chalk
- 5 *Shadow on the Sand* - Joe Dever & Gary Chalk

- 6 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 7 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 8 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 9 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 10 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 11 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 12 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 13 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 14 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 15 *The Wreckage of Tarns* - Joe Dever & Gary Chalk

The Wreckage of Tarns Gamebooks

- 1 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 2 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 3 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 4 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 5 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 6 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 7 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 8 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 9 *The Wreckage of Tarns* - Joe Dever & Gary Chalk
- 10 *The Wreckage of Tarns* - Joe Dever & Gary Chalk

Lone Wolf of the Sea

- 1 *Shipwreck of the Sea* - Joe Dever & John Grant
- 2 *The Dark, Deep Ocean* - Joe Dever & John Grant

Other

- 1 *The Magnificent Companion* - Joe Dever & Gary Chalk
- 2 *The Lone Wolf Books: Putting Back* - Gary Chalk



Contributions



So you have written a program? You want it published?
Follow these guidelines when you send it in to us!

Your *Commodore* is always on the lookout for new programs, hints, tips, articles and more regular content. In fact, if you have something that you think could be of use to other Commodore owners, we want to hear about it.

Below, you will find a list of guidelines that will help us to deal with any data you send us. We don't expect everybody to be building 'William Shakespeares', although if you do follow these simple rules it will make our job a lot easier. (And speedier!)

We are constantly striving to make our magazines more professional in their approach. Therefore, it is to this end that we are making some alterations to how programs should be submitted with a view to publication.

In future, whenever you submit programs or articles for possible publication, we ask you to consult it as submitted according to the guidelines set out below. Please note -

Any programs NOT submitted along these guidelines will automatically be ignored.

Submission Guidelines

1) If possible all material sent to the magazine should be typed or printed out on a computer printer.

2) All text should be double spaced or there should be a blank line between each line of text. There should be a margin of at least 10-15 spaces around the text.

3) The very first page should consist of the following:

Name of the article/program
 Machine that it is for
 Any tapes needed - drive/program, etc
 Your name, address and telephone number

4) The top of every page should have the following information on a Program title
 Your name
 Page number

For example

Example: A Buggy (Page 1)

5) Do not make additional marks on your text, especially underlining.

6) On the bottom of each page put the word **MORE** if there are more pages or **END** if it is the last one.

7) If possible include a listing of your program. If written in machine code then a listing of the source code would be appreciated.

8) Programs should be on either tape or disk with TWO copies included. If problems appear then we have more chance of successful loads. If the program is less than 10 lines it can be included in the text.

9) If your article needs any artwork then supply clear examples of what is needed. We don't expect you to be a first class artist but we do need to see what is required.

An Introduction To Plus/4 Machine Language

Mark Everingham reveals the inner workings of his Plus/4

In future installments of *FourGround*, we shall be printing many articles and programs which need a basic understanding of Machine Language programming. For this reason, over the next few months *FourGround* presents a brief beginners course to machine language. It is obviously not possible to present a full discussion of the subject within the limited space of the magazine, but this course aims to provide a foundation on which to build more advanced knowledge.

For many people, the phrase "machine-code" has horrible images attached to it - strings of binary, incomprehensible letters and digits, level mathematics. However, the facts themselves have about machine-code are usually unfounded - the language can be easy and extremely rewarding to use. If you have tried to learn machine-code before and given up in frustration - don't run off now, most books on the subject tend to be extremely badly written, full of jargon and irrelevant information. This short course has been written to be compact and easy to understand and which retaining all the essential information. For those who can already program in machine language, it should serve as a useful reference work.

What Is Machine-Code?

"Machine-code", "Machine Language" or "Assembly Language" is a computer language just like any other: Basic, PASCAL or C. Machine Language is the native language of the Plus/4 and in fact the computer as its most basic level cannot understand anything else. Basic may seem like the computer's natural language, but it really nothing more than a program written in machine language. At the heart of your Plus/4 is a silicon chip called the 7501 Central Processing Unit (CPU). When you write programs in machine language, you are sending instructions directly to this chip. When you write in Basic, your Basic programs have to be converted into machine-code as they are run, because the 7501 cannot understand them in their Basic form.

So why program in machine language? The reason most people use Basic is because it is so easy to learn. For example, to draw a circle in Basic takes just one command. In machine language - hundreds of individual commands are needed.

This might make Basic seem far more attractive, but the problem with Basic is that it runs incredibly slowly. Machine language runs commands at up to a million instructions per second, whereas with Basic you're lucky if you achieve 60 per second. Another advantage of machine language is that because you're writing at such close proximity to the actual hardware of the machine, you have the ultimate in power over every individual function that the Plus/4 hardware can perform. Once you have begun programming in machine language, Basic will, I can assure you, seem hardly different.

Number Bases - Decimal, Binary, and the Dreaded Hex!

Computers are number machines - Your Plus/4 computer knows nothing about anything but numbers. Everything it does consists of numbers, even the commands it executes are stored as numbers. We human-beings tend to use the number base called decimal or denary, simply because we happen to have ten fingers on our hands, but computers don't like decimal, in fact they use "Hexadecimal" or "Binary".

The concept of number bases is really simple. Think of a decimal number, say 12345. The key to understanding the magnitude of the number is column headings. Look at the number 12345 written under column headings below:

Heading:	10000	1000	100	10	1
Value:	1	2	3	4	5

In order to work out what number is written down, we just add up the products of the values and headings: $1 \times 10000 + 2 \times 1000 + 3 \times 100 + 4 \times 10 + 5$. Each column heading is a power of 10, so the first one is $1, (10^4/10)$, the second one 10×10 , the third $10 \times 10 \times 10$ and so on. The principle for hex and binary is identical, except that for hex, the base is 16 and for binary it is 2. This means that the column headings are 1, 16, 16×16 etc. for hex, and 1, 2, 2×2 etc. for binary. To convert a number to decimal, just repeat the process above using the new column headings. For binary, this is easy, as each column can only hold a number 0 or 1. The number 130 is shown here -

Heading:	128	64	32	16	8	4	2	1
Value:	1	0	1	0	1	0	1	0

The final value is calculated as $16128 + 8x32 + 1x3 + 1x2$. The same method is applied for hexadecimal, except that if the base is 16, each column needs to hold a number between 0 and 15, whereas in decimal only between 0 and 9. This means we need 6 more digits. For these we use the first six letters of the alphabet: A=10, B=11, C=12, D=13, E=14, F=15. The number 4652 is shown below:

Hexing	4096	256	16	1
Value	1	2	3	4

Of course, if in a program we used some numbers which were binary, some decimal and some hex, we wouldn't tell the difference - the number 101 could be 101 decimal, 3 binary or 237 hex. The answer is that binary numbers are preceded by a "B" (percent) sign, hex by a "H" (Strong/ Dollar) sign, and decimal numbers have no prefix. This avoids all possibility of confusion. For most of the time, however, you will find you are hexadecimal, so the problem and need arise.

So why are we not hex and binary instead of decimal? Computers are electronic devices, made up of banks of thousands of switches. A switch can only be on or off, and these settings can be effectively represented by 1 and 0. These 1's or 0's are called bits. Binary digits referred to by their column headings 0.5 (right-half), and collections of eight of these (256 decimal) are called Bytes. Hexadecimal is used because it is much shorter to write down, while retaining a similarity to binary. The number 255 in decimal needs three digits, in binary, 8, but in hex it needs only two:

The Importance Of Numbers

So far we have only talked about numbers for number sake, and this is the natural error which most machine-code users make: in machine language, nothing but numbers are used, but the important thing to remember is that a number can mean anything in real life, numbers can be house numbers, telephone numbers, prices, times, or a host of other things. Machine language is identical, numbers can be colours, sounds, commands, speeds or anything you wish. All of the commands described in the section "The Instructions Set" perform operations on numbers. If you can remember that these numbers can be of any significance, machine-code programming will be elementary!

The Memory Machine

Your Plus/4 is described as a 64K computer. This means that it has enough memory chips to store 65536 byte long binary numbers. You might expect 64x1024, but 65536 is equivalent to 64x1024. 1024 being very rough to 1024 which is a more binary number. Because of the binary system, each of these 65536 "locations" can hold a number in the range 0-255, and each location is referred to by an "address". This is simply a number between 0 and 65535 which points to a byte of memory. In addition, your Plus/4 has two types of memory: ROM (Read Only Memory) and RAM (Random Access Memory). The ROM holds the machine-code program from BASIC. It lies at address 58000-5FFFF, and cannot be changed. The address space 58000-5FFFF is the RAM, although the extra 32K is found at 58000-5FFFF, under the ROM. For the moment, we will only consider the lower 32K.

Each byte of RAM can be put to a different use. To find out which byte of RAM is used for what, you should

consult a manual such as ANCON "CIB-PLUS/4 REFERENCE BOOK" which has a full list. Many locations are used for things like the screen, basic calculations or basic programs. When you write a program in Basic, the Basic program in ROM finds a place in RAM to store the program and puts it there. In machine language, you have to tell the Plus/4 where to put the program, and you must take care not to put it in a place which is used by another part of the computer!

The Tedmon Machine-Code Monitor

On almost all computers, writing a program in machine-code means that you have to go out and buy a piece of software called an Assembler, mainly for you, the Plus/4 has a simple one built in, called TEDMON.

Machine-code programs are written like Basic as a series of text commands such as "TRN", "LDA 38000,X" and "RRR". Each command is represented by a three-letter command, or abbreviation. To actually run these commands, they must be converted into a series of numbers and stored in RAM. This process is called *Assembling*.

The TEDMON program on ROM is your Plus/4 will carry out the simple assembly process for you to save you doing all the work. To assemble a program, type "MON/OUT" from Basic. In order to assemble a program, you have to tell Basic where to put the program. Consult a memory map if you have one to find a suitable place, but for most of the time, a good place to put programs is at 58000. This location in memory leaves you with 32 free for Basic even with a program screen in operation, leaving 32K for your machine-code program - an amount you are not likely to use up for a long, long time! The TEDMON assemble command takes the below form:

A *Source* *was* *opened*

The "A" is the actual assemble command, and can be replaced by a "T" (typically if you prefer. "Source" is the address at which you wish the list of machine-code to be stored. It should always be expressed in hex and preceded by a dollar symbol "\$" as the three-character mnemonic of the command you wish to be assembled, and "opened" is the optional or parameter the command takes. When you enter a hex and path (Source), the hex is stored in memory, and a new address is printed, allowing you to enter the next line. All numbers entered into TEDMON should be hexadecimal, and preceded by a dollar symbol.

When you have finished entering a program, delete the address on the screen. You can exit to Basic by pressing "N" and (Return) for exit, or you can run your program by typing "C" plus the hex start address of your program. The dollar symbol should not be entered.

Once your program is in memory you can look at it or edit it by typing "ED" plus the address at which you want to start looking. A series of lines will be displayed on the screen. This is called *Disassembly*, and the lines on the screen are then to be edited and re-compiled using the normal Screen Editor functions.

We have not covered here every function of TEDMON, but only the most important ones. Full coverage of the TEDMON monitor can be found in your Plus/4 manual.

7501 Processor Registers

As I have said, all of machine-code's commands operate

on numbers. These numbers can be held in three places: in a memory location, in the memory directly after the machine-code command, or in one of the 7501 registers, which are like fast, internal types of memory.

The Program Counter

The program counter (PC) is the only 7501 16-bit register. It is it stored the address where the next command to be executed can be found. When an instruction is executed, the PC is automatically incremented so that it points to the address of the next instruction. When you direct control to another location using a JMP, JSR or branch instruction (see "The 7501 Instruction Set"), the address in the PC is changed accordingly, so you don't have to worry about it.

The Status Register

The status register (SR) is an 8-bit (byte) register which is used to indicate the status of various aspects of the 7501 CPU. In fact, only seven of the eight bits are actually used, and the functions of these bits are discussed below. Each bit is known as a "flag."

Bit 7 - The N (Negative) Flag. The negative or N bit is set to 1 when the last numeric result had bit 7 set, and is reset to 0 when the number had bit 7 cleared. Effectively, numbers over 127 set the N bit, and those below clear it. If bit 7 is used to show the sign of the number, the N bit reflects the sign accordingly.

Bit 6 - The V (Overflow) Flag. The Overflow (V) bit is set to 1 when the last operation resulted in what is known as a two's complement overflow. This is only of use in signed arithmetic, where bit 7 represents the sign (0=+, 8=-). Overflow is set when the sums of bit seven has changed incorrectly, in such cases as two numbers yielding a result greater than will fit in one signed byte.

Bit 4 - The B (Break) Flag. The Break bit is set to 1 when the last operation was a "BRK", or a cleared if the operation was anything else.

Bit 3 - The D (Decimal) Flag. Setting the decimal bit enters the decimal mode of the 7501 processor. Clearing it returns to binary arithmetic. For information on the decimal mode, see the SED instruction.

Bit 2 - The I (Interrupt) Flag. Setting the interrupt (I) bit to one causes interrupts to be disabled, or clearing enables interrupts. The function only works with IRQ (Interrupt Request) interrupts and is covered fully in the section on interrupts.

Bit 1 - The Z (Zero) Flag. The zero bit is set to 1 when the result of the last operation was zero, for instance when subtracting a number from itself. If the result was any number but zero, the zero bit is cleared.

Bit 0 - The C (Carry) Flag. The carry bit is set to 1 when the last addition gave a result which could not be expressed in 8 bits. When subtracting, it is cleared if a borrow was

of eight bits was required. The bit is also used by the shift and rotate instructions, where the last bit is shifted into the carry flag.

The Accumulator (A)

The accumulator, or A register is the most important of the 7501 registers. It is an 8-bit register which is used for all arithmetic and bit manipulation operations, as well as being used for most data transfer functions.

The X Index Register (X)

The X register is one of a pair of "index" registers. It is like the accumulator, in that it is an 8-bit register, but only the most simple operations can be carried out on X. Its primary use is as an index for indexed addressing modes - See the section on addressing modes.

The Y Index Register (Y)

The Y register is identical to the X register, also being used mainly as an index.

The Stack Pointer (SP)

The stack pointer is another 8-bit register, which contains a pointer to the current byte on the 7501 machine stack. See the section which follows on the stack for an overview of its use.

The 7501 Processor Stack

In Basic, you have two constraints to alter the flow of a program: GOTO and GOSUB is simple - it just performs an absolute, unconditional jump to a line. GOSUB does the same thing except that before the jump is performed, the statement to which the program should return is first pushed onto an area of memory called the Basic stack. Then, when a RETURN statement is encountered, the program pulls the return statement back off the Basic stack and jumps to it. Machine language works in the same way, using its own machine stack, except that the GOSUB command is replaced by JSR (Jump to Sub-Routine). When JSR is executed, the address of the next command to be pushed onto the stack is that when a subsequent RTS (Return from Subroutine) is found, the address can be pulled back off the stack and transferred to the PC to continue program execution. The process of pushing and pulling is automatic and controlled by the Stack Pointer. The machine stack is in fact a 256-byte (1 page) area of memory located at \$0000 onwards.

In addition to being used for subroutines, the stack can also be used as a temporary store for numbers, awaiting the need to use a specific area of memory. The PHA, PLA, PHP, PLP instructions push numbers onto the stack and pull them back off it again. When a number is pushed or pulled, the stack pointer is decremented or incremented accordingly. You can however control the stack pointer directly using the TSS and TSS instructions. The stack pointer is a single 8-bit index into the stack area at \$0000.

Note that when using the stack, you need of course match every JSR with an equivalent RTS, and push with equivalent pop. The stack size accommodates 256 bytes, so up to 128 levels of nested sub-routines are theoretically possible. It is extremely unlikely that you will ever need more than one.

Why do I need to use left, right, left, but, but, pop if that's all in the current sub-routine?



One, particularly if one is used to the Burger Museum style of Boarding, is that what you've been brought up on the property would fit in much better. For this is a Boarding for the 90's. The intention is that by using modernizing techniques, that make more realistic (which makes them all the more attractive and the Boarding agent and the Boarding agent's Boarding of the past, they think that there is an effect over the agent is a different. Many people, as they think it is "This is an American Adventure," for far adults that have not been used to it. Many people have been the great of Michael G. Wilson and Edward M. Wilson, respectively, and the John M. Wilson, respectively.

Downloaded by [129.11.24.10] at 17:00 04 March 2016

Food and Drug Administration pulled out of all steps in testing the vaccine to coincide with the center closing on the 15th. The vaccine by Danish pharmaceutical firm, Biontech, is the first that has been shown to have a 100% fatality rate in mice. The vaccine, developed by Dr. David Ho, will be marketed as soon as all human studies are completed.

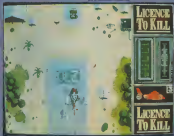
The game is played by two players, one of whom chooses a strategy from a set of strategies, and the other chooses a strategy from a set of strategies.



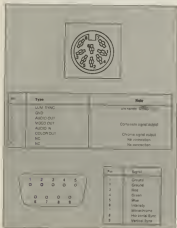
The **F** James
 being a **h** as low
 Lay **C** the **h** but
 gun **o** to a **h**
 Prunder **h** **C** **o**
 low (as **h** off **o** **F**
 hunder but you **h** to **h**
 plant of the **h** first
 our way through a **h**
 an exploding oil barrel and
 we must get back to **h**
 Then it's back in the air to **h**
 have two a plane. Here you **h**
 along the back of the plane where
 must attach a towrope and **h**
 500 feet in.

In home Two 607 men fit a brownie drop, but first he activated the water alarm with a bang but a knife against others' exposed heads. However, he let drawing the Drugs cabinet out of the room, took out the bag, and then returned via Harpoon to replace the pants for a high price but lost watercraft chair, several cartridges, and a new year to the woman and man control.

In the final volume you are in the presence of "Some Key Questions and a



128 Corner



Dear 128 Corner:

Yes, I love a Commodore 128 and have done since I upgraded from the good old 64 back in 1986. I was pleased to see that you have decided to provide a forum in your magazine for this wonderful machine.

I agree with your views regarding the poor availability of software. I suspect the reason for this is that the machine is too small to be a fully

fledged business machine and too large to be a game machine. However, by securing the magazine it is possible to get hold of some significant software that is specific to the computer.

I was surprised that you named the C128 version of *GEORGETTE WORDPRO* as an example of a reasonable word-processor. I was initially put off by the *GEORGETTE*

because it did not allow printing in NLQ mode. The fonts that were provided were too "busy" on my Saw 5000-C for stress letters. This has been solved in *GEORGETTE WORDPRO*.

The biggest advantage of the program is that you can take large sections of *Signus* from *GEORGETTE 128* and paste them into letters without having to write them out again.

For straightforward word-processing I would think that *Super script 128* is best. In some programs *Superbase 128* is another program worth looking out if you want to create a database for my means. The Basic like programming language, included in the program, allows you to manipulate the data in your own applications programs and adds a new dimension to using a database.

The hype surrounding the *Amiga* makes everyone wonder if they shouldn't part with their old computer and buy a new one. I like playing C64 games on the 128 and am happy with spreadsheets, databases and word processor applications that I have for the machine. In the words of the Americans, if it ain't bust, don't fix it.

Colin Morris, Bolton, Lancs

Have you used Fine Master 128 or Paperclip II, both are very excellent applications. I totally agree with your comments about the 128. I too would like my own personal Amiga, but why buy a Amiga to learn to draw when a Master Mouse will do the job?

Dear 128 Corner,

It's good to see that you intend to continue covering the C128 computer. I've had one for a few years now and use it quite regularly for wordprocessing and filing. I must say that I prefer using the C128 over the "real" computers that we have at the office, it's a

The Batman: the Dark Knight and Caped Crusader – call him what you will, his fictional persona has played a major role in our collective mind: his conception in a 1939 issue of Detective Comics. Fifty years on, the twisted vigilante, cloaked and hooded in the dark shadows of a fake New York metropolis known as Gotham City, is about to embark on his second big screen adventure. And this time it's for real.

It is ten years since Warner Brothers, movie-making parent company of Batman's publisher DC Comics first considered the idea of a new Batman feature film. Since that time, names like Wachowski and The Dark Knight Returns have brought a new life to the graphic story genre, adding political, sexual and racial overtones to a myth previously associated with child-bested, costume-clad breakers of nature. And Batman: The Movie is suddenly by no means unlike the Caped Crusader's previous screen outings. Sam Raimi's original 1989 screenplay

original offspring, he is a sharp, not diplomatic, however striking. For the Gold parents, rather than the idealized Adams West West part of the camp and-Strauss is a dream. This trust, POW and KAP, will undoubtedly be a move that pulls the curtain back.

For this reason, Bateman tracks—on film and stage—looked on as a triumph. Her swift rise and equally swift fall? Bateman then Adam Worth's Time, no-one will be left wishing of the promise of her arch-enemy the John. And we all know that Bateman's Jewish nickname Rahab; it's even in the picture, due mainly to the ignorance of the part of the John which followed Jack Nicholson's reluctant acceptance of the role ("I like God created him for the role," producer Mike Upton, quoted as saying). Bateman's director Tim



PC Corner



Our regular *corner* for *PC* products.

I have come to rely on my computer a great deal in the course of my day-to-day life. I have those machines, all of which are made by different manufacturers, all are clones of the IBM PC design and most of them were built by IBM. I use these computers to write software, to do graphics and text for manuals and adverts, control my recording studio, keep up to date with various things and people across three continents and do my financial records in order for my clients of the day and the man I sometimes play cards on over the phone. Before the advent of computers, no man could do this, I suppose.

For prices ranging from \$1000 to \$10,000, these machines are flexible. This adds part due to the availability of a great number of different expansion and expansion cards and the sheer volume of software available. The greater number of manufacturers in the PC market means that there is a wide range of prices and power of the hardware. The wide range of expansion cards available means that you can use a PC to perform almost

any task by simply adding a relatively simple (and therefore cheap) printed circuit board to the computer. As with the hardware, having lots of software for the machine gives you the advantage that computers in any field will give you—quantity and diversity of products.

Another important aspect of the strength of the IBM PC market is that it will have a long life, since IBM with the release of their PS/2 range of computers has set the standards to get a computer designed to the auxiliary and interactive with the PC field. It's sure to know that when you buy a PC that next month or next year you won't be left out on a limb with a computer that has no support or new software.

The speed at which prices are coming down is also amazing. For instance you can now buy a powerful 386-based XT style PC from Atari for under \$2000 with 1 megabyte of RAM and a 10-megabyte disk for around £1000, even though it's what I paid for my 286/50 XT/PC two years ago. There is also a pocket-sized PC being released by Atari soon, which

uses memory cards instead of disks and has a serial port for RS232 communications and a top-top PC from Yamaha designed for professional music applications.

Words for the Whys

The IBM PC is basically a 'versatile' computer, it doesn't have the design compromises that are a feature of a lot of home computers. Having computers keep their prices down and add with a particular goal in mind, they mean that they tend to be the one graphics or extremely low-cost or fast word-processor desktop.

The upshot of this is that the basic PC cannot perform any particular task extremely well, but can perform many tasks competently. The success of the PC design is really quite obvious, but look at the huge number of machines that expansion cards for the design, while a lot these are found in the business or educational market there are a lot of home PCs aimed specifically at the business or home user or can be used for anything.

Possibly one reason for this success is the mechanical integrity of the PC system. While there is nothing particularly novel about the idea of having additional cards for a personal home computer, the allowing of space inside the case for them means that the expanded system is robust enough to stand host to conversion (for example those involving jets and small children). The inclusion of the expansion slot in the PC's design also gives an insight into the intentions of the PC which in turn explains just why there is so much essential and public domain software available for the PC.

The History Bits

The disk operating system (or DOS) used by the PC is based on one of the earliest systems designed for 1-bit microcomputer systems. This DOS was called CP/M, which stands for Control Program for Microcomputers, the major feature of the software was that it wasn't tied to any one computer. This meant that computer manufacturers and even hobbyists could build a computer for which there was a large amount of common software available. A number of popular PC packages were originally written for CP/M such as Wordstar, dBase II and SuperCalc.

The main problem with CP/M was that it was plagued by the fact that there was no standard disk format, the main UK CP/M user group provides disks from its software library in over 120 different disk formats!

The Public Domain

The hobbyist involvement meant that there were a lot of dedicated technical people writing programs for their own use and/or amusement. These people got together on computer clubs in the USA and other parts of the world and released their software in the 'Public Domain' (or PEOs). These programmers were quite happy to let other people use their software and would often provide general support, acting in an informal manner. This informal but, nevertheless, the CP/M operating system was developed by the Amateur PC Users Association (APCUM) which was a group of CP/M users.

The original IBM PC's operating system was very closely related to an extension of CP/M, nicknamed two-disk, the fact was that people who wanted to upgrade their computer to a 16-bit processor could move to a

fairly familiar operating and software environment. The second result was that existing commercial packages could be easily converted (or ported) to the new machine, and since Microsoft, the suppliers of the PC's operating system, also released the DOS software as a separate product other manufacturers could also build it in to their computers.

The User Groups that distributed the CP/M public domain software now also have MS-DOS programs in their catalogues which range from word-processors, games and CAD systems to the entire King James version of the Bible (in fact). The sort of software available from the user groups can be very variable in quality, some of it is very good, some obviously not finished. I use a public domain communications package called Procomm, since it is superior to most commercial terminal emulation programs I've seen.

Operating System Enhancements

Most you, you don't have to put up with the MS-DOS operating system, if you don't want to, there are various ways of 'improving' it. Like the PC hardware, the operating system can be enhanced by adding on what is sometimes called a 'shell-and' Examples of these shells used to improve the user interface of MS-DOS are the graphic environment managers such as Digital Research's GEM or Microsoft Windows and the menu driven systems such as Exec and EDOR.

These programs alter the look and feel of the operating system to make it easier to use either by implementing the WIMP standard (Windows, Icons, Mouse Pointing device) or by displaying the disk files in your current directory on the screen with a menu of standard commands. Microsoft Windows also adds the category of programs that put your PC in ability to run more than one program at a time, the so-called multi-tasking. DOS, Windows and Exec use the Logical Drive (LDR) and Device Drivers.

The End PC

The software has now reached an earlier of the IBM PC compatible computer world. The idea to give an image of things as they really are, copied into the image, into computer has become so popular. In recent reference I hope to go into greater depth on how

to get the PC to do various things, some of which it was not designed to do. I shall also look at some specific programs and expansion cards that I have come across and actually used, not so much as a review, but as a user's comment and as solutions to specific problems which I might have some ideas.

Organisations & Products Mentioned

This section is for reference, the prices are (where shown) exclusive of VAT and aren't necessarily the cheapest, they're just there as a guide.

IBM AT/XT, PC/AT with 80286, 40 Mbyte Hard Disk, 1 Mbyte RAM
Price - £1295,
Supplier - New Computing, 01-229 8410

YAMAHA CI - LCD Laptop PC/AT, 20 Mbyte Hard Disk, 2 Mbyte RAM
Price - £2995,
Supplier - YAMAHA Sales, Condon Street, London

CP/M & MS-DOS User Group, 72 Mill Road, Reading, Hartford, BA3 7LS

CompuLink User Group, Suite 2, The Sanctuary, Surbiton, KT8 6DG, 01-893 84

Fineman v 1.42, communications software
Supplier - CompuLink User Group

Microsoft Windows, Graphics environment for PC
Price - £89,
Supplier - PW Computer Supplies, 01 828 5544

**Exec, DOS file and directory manager from Executive Systems Inc, Price - £44,
Supplier - PW Computer Supplies**

**EDOR, DOS and directory manager from Digital Inc, Price - £44,
Supplier - PW Computer Supplies**

**EDOR, DOS and directory manager from Digital Inc, Price - £44,
Supplier - PW Computer Supplies**

**Windows 1.0, DOS and directory manager from Microsoft, Price - £89,
Supplier - CompuLink User Group, 01-893 84**

Have

VOL 8

COMMODORE



delivered to your door FREE !*

That's right, if you take out a year's subscription to Your Commodore we will make sure that it is delivered to your door each month at no extra charge*. Just fill in the coupon below and send it to the address given with a cheque, money order or credit card instructions to cover the cost of the subscription. We'll do the rest.

* Domestic subscription rates include postage

UK	£13.00
EUROPE	£23.00
MIDDLE EAST	£23.50
FAR EAST	£26.00
REST OF THE WORLD	£24.00
or LIRA	942.00

Armed Rates on Request

Send this form with your remittance to:
INFOWET LTD
 3 River Park Estate
 Elliot Lane,
 BERNHAMSTEAD,
 Herts. HP14 1PL,
 United Kingdom

Please complete my subscription to Your Commodore with the _____ most
 I enclose my cheque/money order for £ _____ made payable to
ARGUS SPECIALIST PUBLICATIONS

Or Debit my Access/Visa number

Valid from _____ to _____ Signature _____

Name _____ Address _____

Postcode _____

☐ New subscriber ☐ Subscription renewal/extension (delete if applicable)

Oops!



Make your programming easier with this package of extra commands for the dedicated Plus/4 user

By Mark Everingham

There are two problems with computers. The first is operational difficulties, and the second is human fallibility. If you are thinking to yourself "What the hell does this mean?" then don't worry — it's really very simple.

Temporal Distortion is the phenomenon by which time seems to become compressed in the presence of a computer. What this means in practice is that when you log up to your Plus/4 for "five minutes before tea", you find that ten has long gone, and that it's about time you were thinking about some breakfast. The answer to this is, of course, a computer alarm-clock, so OOPS! includes such a clock with advanced user-definable alarms and reminders so that you don't forget lunch or that vital doctor's appointment this afternoon.

The other problem with computers which I mentioned before is that although computer calculations

and disk drives are relatively reliable (Commodore's are notorious for being very reliable but equally slow), human beings are really not so dependable. It is all too easy to mistype your Plus/4 or NEW file and only copy of your latest million-dollar trade newspaper. This event will send most rational human beings into a cry of OOPS! (Or similar four-letter words).

In order to protect you from these untoward circumstances, OOPS! has an asNEW-type command and a unique automatic program-backup facility making it as difficult as possible to erase or lose an important program.

The features that OOPS! offers are now becoming more and more common as parts of programming languages as the latest generation of 16-bit microcomputers such as the Commodore 64s, but have not until now been available for the BASIC users of 8-bit home-computers like the Commodore Plus/4.

The OOPS! Basic Commands

OOPS! is a practice control of a program which adds thirty new commands to the internal Commodore Basic Operating-System. This may seem a bit of a daunting prospect at first, but they are all easy to use, and give an idea of how comprehensive the OOPS! system is. The commands can be split into six sections, namely **Clock Commands**, **Alarm Commands**, **Save Commands**, **Printing Commands** and **Mathematical** other commands. In a moment I'll demonstrate the use of each of these, but first a short explanation of command syntax.

Each of the (OOPS!) commands is used in exactly the same way as any standard Commodore Basic commands. Each can be used in Basic Command mode or in a Basic program, and all can be abbreviated and placed later. Because the commands behave like normal Basic commands, any space they have can be replaced with a string variable, for example both the below have exactly the same meaning (to set the 24-hour clock to 12 55 55).

SETCLOCK "12 55 55"

CLS: "12 55 55" SETCLOCK CLS

This property of the commands is useful when writing your own programs. For instance if you can't remember how to work the SETCLOCK command, you could write a program which would set you for the Hours, Minutes and Seconds, and then set the clock for you using something like SETCLOCK "H:" "M:" "S:" with the hours, minutes and seconds stored in H\$, M\$ and S\$ respectively. This facility of the command means that you are not just limited to direct literal commands.

The OOPS! Clock Commands

The OOPS! system includes a 24-hour clock which does not interfere with the normal Basic TZ clock and can be made to appear whenever all the time, whenever else you are doing with your Plus/4 so that you can run programs, print documents, display the disk-directory or whatever you want. The clock will keep running. Below are the commands used to operate the clock.

SETCLOCK "HH/MM/SS"

The **SETCLOCK** command is used to set the current time of the 24-hour clock, as you will obviously not always start using the GOPS! clock at the same time of the day. It's syntax is the time indicated within inverted commas: **HH MM SS** is the time under the 24-hour system. **MM** is the number of minutes and **SS** the number of seconds. Each should be separated using a single colon. The command can yield the error messages shown below which can all be trapped using the normal Basic **TRAP** command, and are displayed as normal.

Syntax Error - You have omitted the time from the command.

Type Mismatch Error - You have typed a numeric parameter instead of the time string.

Illegal Quantify Error - You have entered a time string which neither sat of the form "**HH MM SS**" or has a number in it outside the valid range 0-23 for hours or 0-59 for minutes and seconds.

CLOCK ON/OFF

Once you have set the time on the 24-hour clock using the **SETCLOCK** command, you will want to be able to see what time the clock is showing all the time like a normal alarm clock. To do this just type **CLOCK ON**. From then on the clock is always displayed at the top-right corner of the screen. Note that the GOPS! system cannot be used with the high-resolution graphics screen, so the clock is always visible whatever you are doing. However, there may be times when you don't want the clock display to be visible, for example when writing a program that makes use of the full screen. To remove it from the screen type **CLOCK OFF**. Note that although the clock will now be invisible it is still running and will still display the correct time when a subsequent **CLOCK ON** command is executed. The **CLOCK** command gives only one error message: **Syntax Error** indicating that you have left out, or wrongly explained the **ON/OFF** argument.

START and STOP

The **START** and **STOP** commands are used to respectively start and stop the GOPS! clock. Their main use is for accurate setting of the clock. For example if the time were coming up

to 10 o'clock you could use the program below to set the clock very accurately.

```
10 CSTOP
20 SETCLOCK "10:00:00"
30 PRINT "PRESS A KEY WHEN
  TIME IS 10 O'CLOCK"
40 GETKEY $5
50 CSTART
```

Although it is not necessary to stop the clock while setting it, these commands are useful for setting the clock very accurately or using the clock as a timer started by pressing a key like the button on a stopwatch.

The **START** and **STOP** commands require no arguments and so produce no error messages.

The GOPS! Alarm Commands

If you have anything important that you don't want to forget to do during the day, you can use the GOPS! alarm system of the clock to remind you to do this. As well as giving an alarming audible warning, the alarm can also be made to display a message to remind you exactly what the alarm is set for.

SETALARM "HH MM"

The **SETALARM** command is used to set up the time at which the alarm will sound. It works in the same way as the **SETCLOCK** command but the alarm time is expressed in only hours and minutes and not also seconds. The error messages the command gives are identical to those for the **SETCLOCK** command.

WARNING "Alarm Warning Message"

I have mentioned before that as well as making a noise when the alarm time is reached, GOPS! can be made to display a message as a reminder. This message appears inverted and flashing at the top-left corner of the screen to draw attention to itself. The **WARNING** command is used to set up this message for when the alarm goes off. The message should be enclosed in inverted commas and be no more than 25 characters in length. The message may be made up of all alphanumeric characters except control codes such as **RS&ON** and **RS&OFF**. The error messages that the **WARNING** command yields are shown below.

Syntax Error - You have left out the warning message.

Type Mismatch Error - You have entered a numeric argument in place of the warning message.

String Too Long Error - You have entered a warning message that is more than 25 characters in length.

ALARM ON/OFF

Now that you have set up the alarm time and message, you can switch the alarm on using **ALARM ON**. If you wish to turn the alarm off to meet the time and message, use the **ALARM OFF** command. When the alarm time is reached, a warning sound is made and the warning message displayed on the screen. There are three ways of turning the alarm off. The first way is by typing **ALARM OFF**. The other two methods are described as the **ON** command switches listed under Events. The **ALARM** command gives one error message:

Syntax Error - You have omitted the **ON/OFF** argument.

The GOPS! Event Commands

What are events? Well they are just what they sound like - events, occurrences or happenings during the day. Still more the what? Well, most of us find that we have more than one thing that we want to remember to do on one day, or a single alarm is of limited use. For this reason, GOPS! has eight "Event Alarms". Each of these events works in the same way as the main GOPS! alarm and may be programmed with an event time and associated message. Within a single day you can get your Plan/4 in mind of you of up to eight events, enough for those of you with the most densely packed social diaries!

SETEVENT EVENT #, "HHMM"

The **SETEVENT** command works in the same way as the **SETALARM** command, but is used to set the time at which an event alarm will sound instead of the main GOPS! alarm. Its syntax consists of the number of the event you wish to set, in the range 1-8, followed by the event time enclosed in inverted commas and separated using a comma. The time is entered in an identical format to the **SETALARM**

LABM command The command gives similar error messages, which are shown below.

Syntax ERROR - You have entered the event number, event time or separating comma.

TYPE MISMATCH ERROR - You have entered a string instead of the event number or a number instead of the event time.

ILLEGAL QUANTITY ERROR Either the event number you have specified is beyond the range 1-8, or you have entered an illegal event time.

MESSAGE EVENTS - "Event Message"

As mentioned before, each event alarm can be setup with a corresponding warning message in the same way as the OOPS! main alarm. The MESSAGE command is used to setup one of these messages. It takes as its arguments the event number in the range 1-8 and the message that you want to be displayed when the event time is reached. The message may be up to 35 characters long and should be enclosed in double-quotes and preceded with a comma as shown above. An event message is displayed flashing and inverted in the top-left of the screen in the same manner as the alarm warning, at using the WARNING command. The MESSAGE command gives the error messages that follow:

Syntax ERROR - You have entered the event number, event message or separating comma.

TYPE MISMATCH ERROR - You have replaced the event number with a string or the event message with a number.

ILLEGAL QUANTITY ERROR - The event number you have entered is outside the range 1-8.

STRING TOO LONG ERROR - The event message you have entered is longer than the allowed 35 characters.

ENABLE EV = 1, EV = 2, ..., EV = 8 Once you have setup an event with both event time and event message, you need some way of switching-on or enabling the event, so that when the event time is reached the correct sound is made and message displayed.

This is done by means of the ENABLE command.

The ENABLE command has a flexible syntax so that it can take one or more parameters rather like the Command-line Basic graphics commands. The basic argument is the number of the event that you wish to switch on, or enable. Thus to enable event number 4, just type **ENABLE 4**. If you want to enable more than one event, for instance events 1-8, it is a lot of a pain to have to type **ENABLE 1: ENABLE 2: ENABLE 3** etc. An obvious way can simply type **ENABLE 1,2,3,4** setting down the number of typing needed rarely. Each of the event numbers should be separated from each other using a comma, and lie in the normal range 1-8. Note that the event numbers can be specified in any order, so **ENABLE 1,2,3** has the same effect as **ENABLE 3,2,1**.

When an event has been enabled, at the specified time, the event message will be displayed in the top-left of the screen as explained before, and an alarm will follow. The event alarms are much less harsh, and gentler than the main alarm sound as they are for reminders that are not going to be such a matter of life and death as the main alarm. So if you are of a nervous disposition - use the event alarm instead of the main alarm as they are less of a strain on the heart! The error messages that the ENABLE command yields are shown below.

Syntax ERROR - You have either left out all the event numbers, or finished the line with an unnecessary comma.

TYPE MISMATCH ERROR - You have entered at least one of the event number with a string instead of a number.

ILLEGAL QUANTITY ERROR - At least one of the event numbers you have entered is beyond the allowed range 1-8.

ENABLE EV = 1, EV = 2, ..., EV = 8 The ENABLE command performs exactly the same functions as the ENABLE command, but has the effect of switching-off or disabling an event. Its syntax is identical to that of the ENABLE command and the error messages it gives have the same meanings as for the ENABLE command.

Alarm Priorities and "Shutting Them Up"

Some of you will have realised while reading the explanations of the OOPS! alarm and event commands that it is perfectly possible to not up all eight events and the main alarm to go off at the same time. What would happen if you did that? After all you can't make 9 notes and display 8 messages all at once, can you? The answer is NO: of course not, so a solution is needed in the form of alarm priorities.

Each event and the main alarm has a priority over the others. The alarm comes first, followed by the eight events in numerical order 1-8. To understand this, imagine that you had setup the alarm and events 1,2,3,7 all to sound at 10 o'clock. Because the main alarm has the highest priority, the alarm sound will be made and the alarm message displayed instead of all the others. "OK" you say, but if I can only hear and see the alarm, then what about events 1,2,3 and 7 - they might be important, too! To find out, read on.

OK (CTRL-O)

When an alarm or event goes off, there are several ways you can turn it off.

- (a) If it is the alarm, you can type **ALARM OFF**.
- (b) If you know which event is occurring you can type **DISABLE EVENT**.
- (c) The OK command.

Methods (a) and (b) are all right, but a bit clumsy and, if you don't know which event is sounding, then you will have to try each one of the eight in turn to find out. The solution to this is the OK command. This command simply switches off whatever is sounding at the moment - the main alarm or the current event. Using this command, you don't have to know which event is occurring, and it also turns you quiet a bit at typing.

I discussed earlier the question of what happens when more than one alarm or event is occurring, meaning that you can only see the one with the highest priority - What about the others? Well, when you enter OK, it switches-off the event which is currently displayed on screen, and then displays the next one down the priority scale. For instance imagine that the alarm and all the events were set for the same time. When that time was

enabled, you would see the alarm message. When you type **OK** return the alarm is switched off and then again **!** is displayed. If you type **OK** return again then **OK!** is displayed and so on until all the events have been acknowledged, whereupon the message in the top-left of the screen is erased and the sound turned off again. If you only wanted one alarm, you could use each of the eight events to give you a message of up to 255 characters, using the **OK** command to read the next version of message with all events set for the same time.

For those of you who find typing two characters and pressing **RETURN** a bit tedious, you can achieve the same effect as typing **OK** by pushing **CTRL+Q**.

For those of you who find typing two characters and pressing **RETURN** a bit tedious, you can achieve the same effect as typing **OK** by pushing **CTRL+Q**.

OOPS! Saving Commands

The final important section of OOPS! is its automatic saving facility. Using this function, you can get your Plus4 to automatically save a backup copy of the current program onto disk so that you don't have to retype your copy too fast if you make a disastrous mistake. You can also make OOPS! mark each file it saves so that you have running log of different versions of your program on disk.

SETSAYING MINUTE

As mentioned before, OOPS! can be made to automatically save backup copies of the program that you are working on, so you need to tell OOPS! when it is to save these backup copies. For instance, if you are writing an important program, you may want it saved every five minutes just to make sure that you don't lose it, but, if you are writing a long program, the saving takes a long time so you may want it to be saved every fifteen minutes or half an hour instead. The **SETSAYING** command is used to specify how often OOPS! should save backup copies. The only argument is the period of time between successive savings, expressed in minutes. So, if you want your program saved every five minutes, **SETSAYING 5** will set this up. You can use any time period between every one minute and every hour (60 minutes). The **SETSAYING** error messages are as follows:

Syntax ERROR - You have omitted the **MINUTE** argument.

TYPE MISMATCH ERROR - You

have replaced the **MINUTE** argument with a string.

ILLEGAL QUANTITY ERROR

The time period you have specified in minutes is beyond the range 1-60.

DEVICE 00?

For those of you lucky enough to have more than one disk-drive attached to your Plus4, OOPS! estimates the ability to cope with this. When you are writing a program, you might want to have a utility disk containing an Assembler or Font Editor etc. in disk-drive 1, and an OOPS! backup disk in disk-drive 2. To do this, you would simply type **DEVICE 2**. The **DEV#** argument being the device number of the disk-drive on which OOPS! could save backup files. The device-number can be in the range 1-11 for two disk-drives, but you cannot use device 1 (The Default) as this is really too slow for automatic saving. The **DEVICE** error messages are shown below.

Syntax ERROR You have omitted the device number argument.

TYPE MISMATCH ERROR You have replaced the device number with a string instead of a number.

ILLEGAL DEVICE NUMBER ERROR - The device number you have specified is outside the valid range 1-11.

FILENAME "Filename"

When a backup copy is saved onto disk, it will obviously need to be given a name on the disk and the **FILENAME** command is used to specify this name. The name argument should be enclosed in inverted commas and may be of up to ten characters in length. It may be made up of any alphanumeric characters but may not include spaces. If you do include a space in the name, the actual name will be truncated by the space character. For instance, entering **FILENAME "HALLO THERE!"** results in a file-name of "HALLO". The **FILENAME** error messages are as follows:

Syntax ERROR - You have failed to enter a file-name.

TYPE MISMATCH ERROR - You have either replaced the file-name argument with a numeric argument or have not enclosed the name in inverted commas.

STRING TOO LONG ERROR - The file name you have entered is longer than ten characters.

OOPS! Backup File Marking

It is a little pointless to keep on saving backup copies of a program onto disk unless you know what each file contains. For example if you were saving backup copies every two minutes and each of them were called simply "PROGRAM", you might have some files which were saved two hours ago when you were just starting the program, and some which had only just been saved. Obviously you need to know which is which! OOPS! offers several ways of marking your programs, so you can just choose for the last backup copy to be created every time a new copy is saved. This section discusses the various commands for using the file-marking facility.

VERSION VER#

The first way of marking a backup copy saved by OOPS! is with a version number indicating at what stage of development that backup copy is. You can tell OOPS! which version number of the program you want the first backup copy to be marked with by using the **VERSION** command with **VER#** being the version number. Thus if you want to start with version number 45 enter **VERSION 45**. OOPS! can cope with version numbers in the range 0-999, and this should be quite sufficient unless you are in the habit of saving 1000 backup copies of programs! The **VERSION** command gives the following error messages:

Syntax ERROR You have omitted or the **VER#** argument.

TYPE MISMATCH ERROR - You have replaced the version number argument with a string.

ILLEGAL QUANTITY ERROR - The version number you have specified is outside the valid range 0-999.

NUMBERMARK

This is the first of the commands to tell OOPS! how to mark backup copies of a program. The **NUMBERMARK** command makes OOPS! add the current version number to the end of the file-name. For example if the file-name were set to "BACKUP" and the version number is 55, the file saved on disk would be under these names:

"BACKUP VSE" "BACKUP VSE" ...
"BACKUP VSE"...

As you can see, GQPSI automatically increments the version number each time it saves a backup copy. In the unlikely event that the version number should exceed 9999, it is not to worry. The **NUMBERWARE** command takes no arguments, so needs no error messages.

TIMEMARK

As an alternative to making backup copies with a version number, you can add the current clock time to the filename instead. This allows you to pinpoint the exact version of a program you want, to the nearest minute. For instance, if you had entered **SETSAVING 14: FILENAME "TEXT" TIMEMARK**, the backup copies saved would have been the names shown below, assuming that saving started at two o'clock on the afternoon.

"TEXT 1400" "TEXT 1410" ...
"TEXT 1430"...

The function of GQPSI is most useful when you are making frequent changes to a program. The **TIMEMARK** command gives no error messages.

UNMARKED

If you don't want the backup copies of your program to be marked with version number at all, you can enter the command **UNMARKED** to disable these facilities. From then on, backup copies will have the same name as the original. The **FILENAME** command, with no prefix. However, there is a liability to this - You cannot have more than one file with the same name on a disk, so after the first backup copy has been saved, all subsequent saves will not result in a file being saved onto the disk. To overcome this problem, see the next section on the **REPLACING ON/OFF** command.

REPLACING ON/OFF

As more and more backup copies of a program are saved on a disk, so the amount of free space remaining gets smaller and smaller until the disk overflows. If you are working on a long program then you may not be able to save more than three copies or so on one disk. The answer to this problem and also to saving backups with no suffix is to save the last backup copy before you save the new one. In this way the

amount of space left on a disk will only decrease by the amount you add to a program between saves. For instance if you had entered **VERBOSION 15 NUMBERMARK FILENAME "GQPSI" REPLACING ON**, the following process would take place:

Save "GQPSI" V1. Enter "GQPSI-V1" and Save "GQPSI" V2.

As you can see, this saves a large amount of space on a disk, but also means that your files are not quite as secure because the last copy has to be erased, and it is also slower than just saving new copies. When the first copy of a program is saved with **ReplACING ON**, GQPSI senses that this is the first save, and so does not attempt to erase a non-existent previous file. **REPLACING** gives only the **SYNTAX ERROR** message, if you enter **ON & OFF**.

SAVING ON/OFF

When you have decided what to call your backup copies, how to mark them and whether or not to erase the last one, you can enable the GQPSI saving function using **SAVING ON**. To switch it off again just use **SAVING OFF**. Once saving is enabled on, when it is time to save a backup copy (as decided using **SETSAVING**) the following happens -

- 1 The last you are entering a record.
- 2 The messages below are displayed and their corresponding actions carried out.
- 3 **ERASING "Last File-name" if REPLACING ON**
Last backup copy is erased if **REPLACING ON**.
- 4 **SAVING TO New File-name"**
- 5 Your BASIC program is saved.
- 6 The Computer returns to GQPSI mode.

As you can see from the above, each time a program is saved the line that you are entering at that time has to be erased. This may seem a bit inconvenient, but when you think about it, the most you can lose is 80 characters of program, whereas if you hadn't saved your program you could have lost a kilobyte of it!

The saving function of GQPSI only works from Basic Command mode (When entering programs). If it is a time to save a backup copy while a program is running, GQPSI simply waits for the next time and then tries again. In this way it does not interfere with the

running of the program.

SAVING gives the **SYNTAX ERROR** message if the **ON & OFF** arguments are left out.

STORE (CTRL-X)

Sometimes, you may want to save extra backup copies of a program separately from the usual saving function. It is annoying to have to wait ten minutes (or whatever!) for the next automatic backup, but you may still want the program to be saved with a random number at the time etc. To save a program at any time, using the marking on, you can simply type **STORE**. This has the same effect as if it were time to save a backup copy automatically. Additionally, the **STORE** command can be used even when automatic saving has been disabled using **SAVING OFF**.

Again, for those who consider typing five characters a major undertaking, you can achieve the same effect by typing **STORE** by simply pressing **CTRL-X**. Yes, I would have liked to see **CTRL-S** too, but Commodore lost us to it, for stopping printing!

GQPSI Information Commands

As you can see, GQPSI has a large number of functions which, when you first run the system, can be more than a little confusing. To help you when you are just starting to use GQPSI, I have included in this program five commands to give you information on all aspects of GQPSI from its own manual to the current settings of the alarm or the automatic saving. I'll now discuss each of these in turn.

COMMANDS

GQPSI has thirty commands in all, and you may find it hard to remember all of them. If this is the situation, just type **COMMANDS**. The result is a heading and a list of all thirty commands in reference. You should then not find it too hard to remember which command does what.

CLINFO

Apart from the **COMMANDS** command, there are also four specialized information commands which cover the clock alarm, events and saving functions of GQPSI. The first of these is **CLINFO**.

On entering **CLINFO**, a list of information about the GQPSI clock is printed on-screen, which is as follows -

TIME HH MM SS - The current time of the clock.
DISPLAY ON/OFF - Whether the clock display is on or off as set up by the **CLOCK ON/OFF** command.
STATUS STARTED/ STOPPED - Whether the clock is running or not, as dictated by the **START** and **STOP** commands.

ALINFO

The **ALINFO** command is similar to the **CLINFO** command, but is used to print information about the GOPS' alarm instead of the clock. It gives the information shown below:

TIME HH MM - The time at which the alarm is set to go off.
STATUS ON/OFF - Whether the alarm is enabled or disabled using the **ALARM ON/OFF** command.
MESSAGE "Warning" - The warning message to be displayed when the alarm time is reached, as defined using the **WARNING** command.

EVINFO

The **EVINFO** command displays a list of information about the current settings for each of GOPS' eight events. All are listed under a set of headings which may be explained as follows -

- The event number to which this line of information refers.
TIME - The time at which that event will occur.
STS - The status of the event - whether or not it is on or off as defined using the **ENABLE** and **DISABLE** commands.
MESSAGE - The event message assigned to this event using the **MESSAGE** command.

Each line shows the same information, but for event number - as shown on the left of the line.

SVINFO

The last of the GOPS' information commands is the **SVINFO** command which is used to display the current settings for the automatic saving function is GOPS'. The list of information the command gives is shown below:

TIMING MM - Period between subsequent saves in minutes as defined by the **SETSAVING** command.
STATUS ON/OFF - Whether or not the automatic saving is on or off, as set up using **SAVING ON/OFF**.
MODE UNMARKED/ NUMBERMARK/ - The current method for marking backup copies.
TIMEMARK
REPLACING ON/OFF - Whether or not the replacing or creating function is enabled by the **REPLACING** command.
VERSION YYYY - The current version number set up using the **VERSION** command.
DEVICE DD - The current saving device-number as set up using the **DEVICE** command.
FILENAME "Filename" - The current save file-name defined with the **FILENAME** command.

Miscellaneous Commands

As well as the specific function commands and information commands, I have provided GOPS' with a few extra commands to make life a bit easier when using the system. This section deals with these three system-
 up commands, namely **WINDOW**, **RECOVER** and **RESET**.

WINDOW NOW

You may have noticed that when the clock display is enabled, scrolling the screen causes the clock to blink in an annoying way. You may also have seen that some strange effects can be achieved using the Delete and Insert keys on the top two lines of the screen. It would be better if all such tasks were in an area of the screen not affected by the clock display or flashing messages, which would remedy the problems outlined above. The answer is the **WINDOW** command.

The **WINDOW** command is used to reduce the visible area of the screen to an area below the clock display so that it does not interfere with any work you're doing. It is the equivalent of moving the cursor to the corners of the area and pressing ESC-T and ESC-B. Two windows are available which are similar to the ESC-N screens, but with the top edge below the clock display. The **WINDOW** command takes only one argument - **WINDOW** B enables the ESC-N equivalent window, and **WINDOW** I is the ESC-R equivalent. These are best explained by trying them out yourselves. The edge coordinates for the respective windows are as follows:

WINDOW O Top-left Corner (8,2)
 Bottom-Left Corner (26,24)

WINDOW I Top-Left Corner (1,3)
 Bottom-Left Corner (18,25)

The **WINDOW** command gives the error messages shown below:

"TEXTX ERROR - You have exceeded the window number argument.

"TYPE MISMATCH ERROR - You have mistakenly replaced the window number with a string.

ILLEGAL QUANTITY ERROR

The window number you have specified is either 0 or 1 as it should be.

RECOVER

In addition to the automatic backup facility of GCPFS to map your long valuable programs, GCPFS also has an **SAVE W-type** command **RECOVER**. If you enter a program and then crash it from memory using the **NEW** command, you can get it back by typing **RECOVER**. The command will of course not work after you press the **RESET** button on your Plus 4, doing this erases the GCPFS program code. Some of the time you may be able to get your program back in this situation by reloading the GCPFS program then typing **RECOVER**. You should be wary to use the command when you've just crashed your computer as, because it will give some funny results with no program data present.

The **RECOVER** command has no arguments so it yields no error messages.

RESET

The final GCPFS command, **RESET**, is used to reset the GCPFS data to its default values. This resets the clock on, resets all the events and brings etc. to the values that they have when you first use GCPFS. These default values are what is typical values such as you might frequently use. These default settings are shown later.

GCPFS Abbreviations

Like the normal Commodore Basic commands, GCPFS commands can be typed abbreviated to save on typing. When typed into a program, abbreviated, they are converted into their full form on being up or just as easy to understand as typing them is. Below is a list of the GCPFS command abbreviations as provided here. All abbreviations take exactly the same syntax as the full command.

GCPFS Kickstart File Generator

You should now have sufficient knowledge to be able to use all of the GCPFS commands competently. I'll now discuss the GCPFS Kickstart File Generator program (Listing 3).

Most of the time when using GCPFS you will find that you always use the same saving setup, or the same event times and messages, so it is a

Command	Argument	Maximum Abbreviation
SETCLOCK	"HH MM SS"	S SHIFT-8
CLOCK	ON/OFF	C SHIFT-L
SETALARM	"HH MM"	SET SHIFT-A
WARNING	"Message"	W SHIFT-A
ALARM	ON/OFF	A SHIFT-L
SETEVENT	EV, "HH MM"	SET SHIFT-E
MESSAGE	EV, "Message"	M SHIFT-E
ENABLE	EV1, EV2	N SHIFT-N
DISABLE	EV1, EV2	O SHIFT-I
OK	-	OK
SETSAVING	SAV	SET SHIFT-S
DEVICE	DEV	D SHIFT-E
FILENAME	"Filename"	F SHIFT-I
VERSION	VER	V SHIFT-E
NUMBERMARK	-	N SHIFT-O
TIMEMARK	-	T SHIFT-I
UNMARKED	-	U SHIFT-N
REPLACING	ON/OFF	R SHIFT-E
SAVING	ON/OFF	S SHIFT-A
STORE	-	S SHIFT-T
CSTART	-	C SHIFT-S
CSTOP	-	CST SHIFT-O
RESET	-	RE SHIFT-S
RECOVER	-	RE SHIFT-C
WINDOW	0, 1	W SHIFT-S
COMMANDS	-	C SHIFT-O
CLINFO	-	CL SHIFT-I
ALINFO	-	AL SHIFT-I
EVINFO	-	E SHIFT-V
SVINFO	-	S SHIFT-V

lot of a waste of time to keep on typing the list of **SETEVENT** commands or whatever to set these values every time you use GCPFS or every time you start your Plus 4. It would be much easier to just write a short "Kickstart" program to set up the GCPFS functions so that all you need to do whenever you use GCPFS is to load and RUN this program. This is of course possible, and you can write such a program like any other Basic program using the GCPFS commands, but as an alternative, I have included the GCPFS Kickstart File Generator which can write such a program itself - A program writing another program!

When you run the Kickstart File Generator (Humbleboth released to us KICKGEN), it will ask you a series of questions. When it has done this, it will create a Kickstart program and then save it to disk or cassette. In this section I'll discuss all the questions KICKGEN asks, with the expected replies shown in bold parentheses.

(1) **WINDOW (0/1)** Enter which of the window you wish to start using GCPFS within. The ESC W type window 0, or the ESC R window 1.

(2) **CLOCK TIME (HH/MM/SS)** Enter the time you wish the GCPFS clock to be set to when you first start using GCPFS. Note that when you are entering times into KICKGEN, you must separate the Hours, Minutes and Seconds with any character EXCEPT a colon. For example 10:00:00, 10-00-00 and 10 00 00 are all valid, but 10:00:00 will not work as the BASIC input routine cannot cope with it.

(3) **CLOCK DISPLAY (ON/OFF)** Enter whether or not you want the clock display enabled ON or OFF when you start using GCPFS.

(4) **DO YOU WANT TO SET THE ALARM? (YES/NO)** If you don't want to bother with setting the GCPFS alarm, just enter NO and go on to section (5). If you do want to set the alarm, follow the steps below.

(5) **ALARM TIME (HH/MM)** Enter the time you want to set the alarm to.

(6) **ALARM WARNING (Message)** Enter the message that you want to be displayed on-screen when the alarm time is reached.

(7) **ALARM STATUS (ON/OFF)**
Enter whether or not you want the alarm to be switched ON or OFF when you start using GOFPS?

(8) **D0 YOU WANT TO SET ANY EVENTS? (YES/NO)** If you don't want any events set up when you start using GOFPS? enter NO and go to step (12). Otherwise, enter YES and follow the steps below

(9) **EVENT X TIME (HH/MM)**
When you have answered that you do want to set some of the GOFPS' events, you will be asked a series of questions for each of the events 1-8. The prompts displayed are of the same form for each event, with X being the event number to which the query refers. E.g. "EVENT 5 TIME" or "EVENT ? STATUS". The first entry, EVENT TIME is the time you want the particular event to be set to, but if you don't want to set that event, just press RETURN without entering anything and go on to the next event

(10) **EVENT X MESSAGE (Message)**
Enter the message you wish to be displayed when the time for event X is reached

(11) **EVENT X STATUS (ON/OFF)**
Enter whether or not you want event number X to be enabled (ON) or disabled (OFF)

(12) **D0 YOU WANT TO SET THE SAVING? (YES/NO)** If you want to set up the automatic saving facility of GOFPS? enter YES and follow the steps below. If not, enter NO and skip to the end of this section.

(13) **SAVE TIMING (0-60)** Enter the period of time (in minutes) between subsequent automatic saves

(14) **DEVICE NUMBER (0-10)** Enter the device number of the disk-drive on which you wish GOFPS? to save its automatic backup copies

(15) **FILENAME (Name)** Enter the filename under which you want backup copies of a program to be saved

(16) **VERSION NUMBER (0-9999)**
Enter the version number that you wish the first backup copy to be marked with when VERSIONMARK is entered

(17) **MARKING MODE (N,T,C)**
Enter the first character of the mode

by which you wish backup copies to be marked. This should be one of (N)umbermark, (T)imestamp, or (C)ommand.

(18) **REPLACING (ON/OFF)** Enter whether or not you wish the previous backup copy to be erased when a new backup copy is saved.

(19) **SAVING STATUS (ON/OFF)**
Enter whether or not you want the automatic saving function of GOFPS? to be switched ON or OFF when you first start using the system

When you have answered all the questions the screen window will clear and the message "COMPILING KICKSTART FILE" will be displayed. After this, program lines will gradually be displayed. When set up all the GOFPS? functions as you have specified by answering KICKGEN's questions. When KICKGEN is displaying program lines, what it is doing is actually building a BASIC program into another area of memory from itself. When it has finished this process, KICKGEN displays the message "COMPILING COMPLETE - TAPS,GOSE (T/GO)" If you want to save the Kickstart program to Tape, press "T" or if to Disk, press "O". KICKGEN then asks you to insert a disk or tape, and then when you press RETURN it will save the Kickstart program onto the tape or disk.

When KICKGEN has saved the Kickstart program, from that time you can run up your preferred GOFPS? functions simply by typing the following

LOAD "KICKSTART" if you are using a disk-drive

LOAD "KICKSTART" if you are using a Datadisc as a more secure media.

When the program has loaded, simply type RUN and hit RETURN. Of course, don't forget to load the GOFPS? main program before!

That completes our discussion of the GOFPS? commands and KICKGEN program. It's now down to the most important aspect of GOFPS? - getting the thing started!

The GOFPS? system comes in two parts which are the GOFPS? Basic Loader (Listing 1) and the GOFPS? Kickstart File Generator (Listing 2). The Basic Loader is simply used to create the machine-code GOFPS? program, and is used as follows:

Enter Listing 1 and RUN. The program will give messages to allow you to correct the data lines. Note that you should start the session of data with line 1000. To help you, you can use the line below to set up automatic line numbering, and function key 1 to produce the DATA associated with our keywords -

KEY1,"DATA" AUTO 10

When you have entered Listing 1 and corrected all your mistakes, the program will ask you whether you want to save the GOFPS? programs to disk or tape. Press "O" or "T" to choose which and then insert a diskette or tape and press RETURN to save the programs. The programs saved are 1st BASIC Loader and (3) GOFPS? Machine-Code Program. When these programs have been saved, to use the GOFPS? system just reset your Plus4 and type the following:

For Tape Users: LOAD "GOFPS?" RETURN RUN RETURN

For Disk Users: LOAD "GOFPS?" RETURN RUN RETURN

When you type RUN, the GOFPS? commands are enabled, a title screen displayed and the clock started on. You can then write programs or use GOFPS? commands. If you use the RUN/STOP-RESET combination to escape from a machine-code program, typing K RETURN from TED-MON will get you back into Basic and re-enable the GOFPS? functions.

As mentioned before, the saving facility of GOFPS? cannot be used with a Datadisc, but all the clock, alarm and event facilities may be used with either disk or tape.

The KICKGEN program (Listing 2) can be entered just like any other normal BASIC program. It can be saved as shown below:

For Tape Users: SAVE "GOFPS? KICKGEN" RETURN

For Disk Users: DSAVE "GOFPS? KICKGEN" RETURN

The GOFPS? KICKGEN program may be used either with or without GOFPS? resident in memory.

Believe it or not, that concludes this article about GOFPS? Now then, it's just format that disk - there we go - bang on, what does that disk-intel say? "GOFPS? Machine disk". Oh, yeah!

100

[illegible][illegible][illegible]

PLUS4 PROGRAM

[illegible][illegible][illegible]

100

PLT54 PROGRAM

[illegible][illegible][illegible]

Logically Speaking



*Get your thinking caps on for this excursion into
Mathematics and Logic*

By Pete Gerrard

I would imagine that many of you have dabbled with the game of Chess at one time or another, or perhaps a simpler game of Draughts on the same board. The complexities of Chess can be baffling for the beginner, the rules of Draughts are most easily understood, but there is less both of them remain fascinating games for expert and novice alike. The topic under discussion now revolves around the board rather than the pieces on it, so, if you have such a board, you may want to put it out and study it. If not, you will just have to imagine an eight by eight board, don't worry about black and white squares, just stick to the eight by eight board, and we can continue.

After a quick calculation you might say that there are 64 squares on the board, and indeed there are 64 squares that are just one square in width and height. But what about the squares that are two squares in width and height, or three squares, or even eight

squares? Of course, there is only one of the latter, but there are many more of the other ones and all the intermediate ones. How many are there in total?

As if that were not puzzling enough, consider also that there are squares, which form rectangles (formed from two or more squares). Rectangles can be two squares by one, or four squares by two, and there are far more of them than are the more conventional squares.

So how many rectangles can be formed on a standard Chess board? And having discovered that one, what is the formula for the combined number of rectangles and squares?

Extremely few as there are no standard formulae for solving non-linear problems. However, if I would simply tell you the answer for a square the size of a chessboard, that would tell you the answer for a two square board, or a four square board, or a sixteen square board, or a thirty-two square board. Because it is less than 1000, in the form of the following program. This will allow you to solve the middle of how many squares and rectangles there are for a whole range of squares from the simple two by two to the much more

complex twenty by twenty.

Of course, you could use even larger numbers, although the square itself would not be drawn on the screen and the listing would need extending accordingly.

First, let us consider the formulae. If we have a variable 'A' equal to the size of the square (and then 'N' would equal 8 for a standard Chess board), then the combined total of squares and rectangles that can be formed is given by the formula $(N^2 + N) * (N + 1) / 6$. This gives us the total number and the only other number of squares is given by the formula $(2 * N^2 + N^2 + 1) / 6$. From this we can of course work out the number of rectangles easily enough, but just for the sake of completeness, we should first tell you the number of rectangles not including squares is $(2 * N^2 + N^2 - 2 * N) / 12$. A complicated one!

This brings us to the program listing, and a simple enough one it is too since we know the relevant formulae. Lines 10 and 20 serve to identify A, while lines 30 and 40 select our colour scheme and clear the screen for answers. Lines 50 to 60 then allow the

can't to input a number in the range 1 to 20, and could be extended if larger ranges were required, although you could date for advanced users to accept digits 0 through 9 and skip straight to line 140.

Line 150 shows the initial again, line 80 to 130 define and draw the square in its entirety, before we get to the actual calculations. After a brief statement on line 140, line 130 then works out the total number of squares and rectangles using the formula given earlier. Line 150 calculates the number of squares and the number of rectangles including squares. Then line 160 calculates the actual value for the correct answer, can now be verified by drawing a trivial square (say of size 1, for instance, which is easily calculated in the tape).

The answers are displayed by line 170 to 180, and line 200 rounds everything off and prevents the problem falling through to line 210 and producing a RETURN WITHOUT GOSUB error.

That is the program, there are the formulas, and no amazing high-mathematical diversion they form as well. However, an article of this nature would perhaps complete without giving you some ideas to do as well. As we have seen, there are formulae for squares, but what about triangles? Imagine an isosceles triangle that is divided into right-angled triangles along its base right down to the running edge, and the bottom line is with triangles of the same size. Rather like an eight by eight chess board, we might be said to have an eight by eight by eight triangle. Reconsidering that triangles can be formed again down to the right way up, what is the total number of triangles contained within the big triangle, giving that the big triangle is a whole with one to the total, just as the chess board adds one to the number of squares. Okay if you know the formula, but if you don't...!!

Switching The Bases

As human beings we are accustomed to counting in a numerical system that uses ten as its base, hence we arrive at what is called a decimal system. This is not particularly convenient for computers which are ideally suited to the so-called binary system.

As far as computers are concerned an electronic circuit can be either

tuned on or off, it only has two possible values associated with it, and, if we give those values the numerical associations of zero and one, we have the basis for the binary system.

Binary and decimal systems are happily thought, and as a side account, they combine into the reckoning. This is because of hexadecimal, using the number 16 as its base. Unfortunately the Arabic numerals have only one own device only, based on the 10, using the symbols 0, 1, 2, and so on. Rather than creating new symbols, hexadecimal uses letters of the alphabet. These letters are A, B, C, D, E and F, and this gives us ten additional numbers of the form 123F, for example.

In order to convert numbers from one numerical system to another, we need only know the base that a being used. For instance, the decimal number 1234 is easily translated as 4 plus 3 times 10 plus 2 times 10 is the power 2 plus 1 times 10 is the power of 3. The further left we go in the number, the greater the power of ten that is being used. To make our earlier mathematical example of 123F, knowing that F is equivalent to the decimal number 14 and F is equivalent to the decimal number 15, we would get 14 plus 15 times 16 plus 3 times 16 is the power of 2 plus 1 times 16 is the power of 3, or 1990 is the same familiar decimal system.

Thus the logic for all converting systems must be the same being used, and by employing that base number used, with appropriate powers by the number concerned, we have easily converted from one numerical base to another. Another example let us assume a numerical system that uses the base 7. An unusual choice perhaps. Hence interesting one nonetheless. If we look at a number of 1316 which we expressed using the base of 7, to convert it back into our own decimal system we have 4 plus 3 times 7 plus 2 times 7 is power 2 plus 1 times 7 is power 3. This equates to a decimal number of 466.

Now we arrive at the mathematical diversion. To begin with, we will be using a base of 8, a base of 8, and a base of 7. In other words, three different bases. What we want to find is a solution to the following equation, assuming the variable 'A' to be the number we are seeking for A (base 8) minus A (base 5) equals A (base 7). The same number, but using a

different numerical base for each stage that it is expressed. What if, however, we want to find a base digit number that will solve the equation, with no possible numbers left behind, 47 words later to find a number which has an second and third digit equal A number like 1321, for example, although that is not the answer.

Many such problems can be solved, and I shall leave you perhaps to ponder on at the end of this article. For now, let us consider the problem as it stands and the program, leaving that matter as to solve it. With the rules of the problem clearly expressed, and the method of translating numbers from one base to another already defined, the being should pose no problems.

Line 10 serves to identify the program and its author before line 20 and the coding begins, while line 30 starts the typing. The loop in line 40 sets up to count the possible four digit numbers, and with line 50 and 60 we set up a string variable to put in one check for the presence of the equal digits on the second and third positions within those four digit numbers.

If things don't go on the same then line 70 sends us straight to line 180 to continue the search, ultimately concluding with that line if no such number is found. However, if the second and third digits are the same then we go on down to lines 110, 120, 130 and 170 in order to convert that number to base 8, 5, 7 and 20 respectively, before going to line 190 and displaying the numbers on the screen to allow the user of the program being made.

If the equation now holds true then line 80 tells us of the relevant number before leaving the program with an END statement, otherwise we carry on with the loop in line 180.

The final number that conforms with our equation is the decimal number 1312. That is, 1312 in base 8 minus 1312 in base 5 equals 1312 in base 7. The decimal number 2664 also obeys the rules.

That is a simple example of the kind of problem that can be set involving a mixture of bases for computer purposes. Now consider a different sort of problem using the equation A (base 8) equals A (base 5) plus A (base 7). If the variable 'A' is 1, then 1 (base 8) equals 1 (base 5) plus 1 (base 7). If the variable 'A' is 2, then 2 (base 8) equals 2 (base 5) plus 2 (base 7). The same number, but using a

Binders

Organize and protect your disk with Commodore Disk User disk binders and data disks.

Why not keep your Commodore Disk User program collection alongside your magazines in a stylish Disk User disk binder? The binder comes complete with 10 disk sleeves to organize and protect your program disks. Why not buy a disk binder to house all of your data disks? We can even supply Commodore Disk User data disks. The Commodore Disk User logo immediately identifies your disks and clears room on the label to enter and document the data details. Send for your disks and binders now!

Prices are as follows:

Commodore Disk User Binder £9.95, including

10 sleeves. Order code **BDYU1**

Commodore Disk User Binder with 10 sleeves and

10 disks. £9.95 Order code **BDYU1**

10 sleeves for insertion in binder. £1.50 Order code

BDSS1

20 sleeves for inclusion in binder. £2.75 Order code

BDSS2

10 Commodore Disk User data disks. £5.95 Order

code **BDON**

All orders should be sent to: **YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 8ST.** Please allow 28 days for delivery.



PRODUCT NAME	ORDER CODE	QUANTITY	PRICE
Overseas postage add £1.00			
			TOTAL



NEXT MONTH

NIGHTBREED

Oliver Barker spits his guts in an exclusive interview about his new film and the game to follow

LIFESTYLES 2000

On the spot report

DESERT ISLAND DISKS

Larry Crowther is the first to contribute to our new regular spot

Plus Role Playing Games, Play By Mail, Software in the Office...

Reasoning On The 128

*We continue with our expert system for the 128
computer*

By Paul Schofield

In the last installment of the series we introduced you to the idea of a simple expert system, from now on we will look at the more sophisticated knowledge base which I introduced in that article. This system comprises of three programmes:

EDR — the source program editor
KBDN — the knowledge base compiler
INQUIRE — the inference engine

Why three programs? The most important consideration is space available for data, which is increased by keeping the programs small. Speed of operation is also required, so when the application has been debugged, it is only necessary to use the **INQUIRE** program which reads in the data in a compressed and programmed format. In this section, we will be concerned mainly with the editor, but first a quick overview of the whole system.

Defining a knowledge base

To create the knowledge base for the reason program, we first processed all the data that we had on the subject in questions and arranged it into a tree structure. In the case of the very simple chemical analysis example, we ended up with a tree simple syntactic tree, which chemists took an different path to any of the solutions. If we were to extend this, we would have to start using trees to identify particular chemicals and on the tree would no longer be syntactic. This is similar to the car maintenance example, where

it was suggested that the first questions might be:

fuel tank is empty?
battery is flat?

In this case, we pose these as starting questions, because they are easy to check and the produce a simple solution.

In terms of our expert system however, they represent a fundamentally different approach. We pose a solution to the problem, namely, put petrol in fuel tank and ask a set of questions to support it. If the answers are affirmative we have an answer, otherwise we take the next possible solution and test this.

This is the approach that our new knowledge base system will adopt. It is probably the most widely used in the type of application and has a number of advantages. There are, however, attendant disadvantages, and as the system must endeavour to take account of these:

Incomplete data

One of the first rules one learns as a programmer is, garbage in means garbage out and this is particularly true of this type of expert system. To define your knowledge base, it is necessary to define your targets (the possible solutions) and the conditions which must be satisfied for a particular target to be the solution. The order at which you define the targets also implicitly defines the order in which they are evaluated. This is good, because at the outset you can start by just entering conditions, which you

know to be true about a target. The disadvantage is fairly obvious by an example. Suppose you write a knowledge base for animal identification. The first target is cat and the conditions are:

Has four legs?
Has a tail?

In use, the knowledge base will tell you that a wide variety of animals are cats. The real message is "according to the supplied criteria a possible solution is a cat". For the expert system to be useful it's not sufficient for it simply to stop at the first solution, it must provide additional options.

Wfs — A justification of the solution.
More — Continue to look for other solutions.

With the aid of these tools it is possible to identify sections of the knowledge base, which are incomplete and identify suitable conditions to apply. We will consider such problems in a little more detail when we consider the source definition language and the inference engine, but this should suffice as a general introduction to the system.

EDR — The source file editor

The obvious question is why not use the Basic editor? One reason of course is that the Basic editor produces untidy program files, but in general it is more convenient to use either a general purpose text editor or one specific to the particular application. **EDR** is specific to this application, but

would not require much modification to make a general purpose text editor. To run the editor, simply enter `run EDIT`. The editor initializes various variables and then displays the main options menu. Most of these are self-explanatory, so we will just consider option 1, create/edit RAL source file. When this option is selected, a help bar is displayed at the top of the screen showing the functions assigned to the function keys. The remainder of the screen is either blank (when) or displays the start of the currently loaded file. The files produced are sequential files, so no line numbers are needed, just enter the text you require. The following special function keys are also supported:

CURSOR	Move up/down/left/
KEYS	right one character
DEL	Delete character left of cursor
INST	Insert blank character at cursor
F1	Insert blank line
F2	Delete current line
RETURN	Move cursor to start of next line
F4	Delete remainder of line to right of cursor
F4	Copy line to paste buffer and delete it
F3	Paste saved line at cur
F5	Move cur to top of file
F7	Move cur to end of file
HOME	Move cur to top left of screen
CTRL/U	Move up 20 lines or to top of file
CTRL/D	Move down 20 lines or to end of file
F8	exit from edit mode

The automatic function of the editor is that when the bottom of the screen is reached, the text is scrolled up 10 lines.

Modifying The Editor

The functions supported by the editor are all that you are likely to need for entering knowledge base definitions. For more general use a few modifications are likely to be needed. New edit functions can be added very easily. Add a test for the key in question between line 10300 and line 10350 and insert the corresponding subroutines somewhere between line 2800 and line 4999.

Two additional modifications may also be required.

1) Currently the number of lines of text is written at the start of the disk file. If this is considered undesirable (the disk read/write routines must be changed to delete tail

2) Commodore Basic has trouble with commas when writing text to disk files unless special precautions are taken. As the knowledge base definitions language does not use commas, this has not been done.

We shall now look at the definitions language RAL and the compiler program EGEN.

Having experimented with the REASON program, and typing in the EDIT Editor. We will now look at the knowledge base definitions. Later, we will look at the compiler program, EGEN. Part of all though, we will look at the definition language. As these 3 programs together represent the package Reason Version 3, I have called the language, Reason Application Language (RAL).

Before going into details though, a quick look at some of the objectives. Although EGEN is technically a compiler, I prefer to think of it as a simple preprocessor. It does not generate machine code, but simply takes a description of data structures in a readable format and translates them into a set of arrays suitable for drawing an inference engine. It was also considered important, that the compiler should be able to output the processed knowledge base on a different disk to the source file. This is administratively very convenient, but, if unknown disk swapping is to be avoided, requires that the source source file is read before attempting to write the output file. This in turn dictates that large string storage requirements are provided and in order not to unduly restrict the size of applications the language becomes a little cryptic in places.

It is possible to change this approach, but I would recommend trying the existing system first. It's not really that difficult.

Structure of RAL

RAL is a block structured language like Pascal or C, but the syntax of the individual blocks is a bit more odd.

Outside any of the blocks comment lines are permitted, so the basic structure of a RAL program is

```
comment
RAL BLOCK
comment
```

```
comment
RAL BLOCK
comment
```

There are 3 types of RAL block -

HELP Initial instructions displayed to user of base
TARGETS Define the set of target relations
CONDITIONS Define the set of questions the user is asked
RELATIONS - Define set of rules for calculating targets
IMPLIES Define how additional conditions may be inferred from a user response

The structure of each block is as follows

```
BLOCK HEADER
Block comment
END [comment]
```

The square brackets [] indicate optional elements

Comments

As mentioned previously, comments may be included anywhere outside a block, they have the format

```
[any text string]
```

It is important that the ';' at the first character on comment lines and those are simply ignored by the compiler. It would not be difficult to permit comments within blocks, but it would greatly slow down the compiler and is not very useful.

Help Section

The help section is optional, but should normally be included in all RAL definitions. It simply defines a number of lines of explanatory text that will be displayed to the user before any questions are asked. It may typically be used to display a title, make area for creating the knowledge base and providing any necessary background information. It is not necessary to include any information on how to respond to various questions as this is automatically output by the INQUIRE program. The format of the help section is

HELP target
help=number lines
END[HELP]

where
target defines the number of lines of help information following.
help=number is the help information in the format to be displayed.

Targets Section

The targets section defines the test arrays describing the target solutions and a *currently mandatory*. The format of the targets section is

TARGETS target
target=lines
END[TARGETS]

where
target defines the number of entries to be reserved for targets.
 and *target=lines* have the format

[*index*] *target=text*

where
index is an optional target number reference (1,2,3 etc.)
target=text is the text that will be displayed to describe the solution.

Note

The purpose of *target* is to define internal storage requirements and it must be the same or greater than the number of following targets. The *index* is purely a comment, a bit like a Basic line number. It is used to refer sections to identify specific targets and is therefore useful in large knowledge bases. The first index is always 1 and they are allocated consecutively.

Conditions Section

The conditions section defines the questions which will be asked to the user and is *obviously mandatory*. The format of the conditions section is

CONDITIONS target
condition=lines
END[CONDITIONS]

where
target defines the number of entries to be reserved for conditions.
condition=lines have the format

[*index*] *condition=text*

where
index is an optional condition number

reference (1,2,3 etc.)
condition=text is the text of the questions displayed to the user.
target and *index* are treated as for targets.

Relations Section

The relations section defines the rules for evaluating targets according to the answers to the conditions questions. The results of the various conditions may be either *conditions* (must be true for target to be a solution) or *relations* (must be false for target to be a solution).
 By default, all conditions are considered *irrelevant unless mentioned explicitly within a rule in the relations section*. The format of the relations section is

RELATIONS
relation=rules
END[RELATIONS]

where
relation rules have the format *operator operand condition index* for
 where
operator is "&" following conditions must be true, or "&" following conditions must be false.
condition=operand=lines is a number of conditions *operands* separated by spaces.

Implies Section

The implies section is optional. It defines the results of further conditions, which may be determined from the result of the current conditions. The object of this section is to avoid asking questions whose results should already be known and to a lesser extent, avoid oversteering an possible targets as quickly as possible. For example, if during the investigation of one target the question is asked "IS OBJECT BLUE?" and the answer is true, it is quite obvious that a later question "IS OBJECT RED?" must have the answer false. The implies section allows upto three additional results to be implied for both a true or false answer to a particular condition. Its format is

IMPLIES
implication=rules
END[IMPLIES]

where the format of *implication rules* is

result=condition *implied=condition*
[implied=condition] *[implied=condition]*

and the format of both *result=condition* and *implied condition* is

"&" *condition=rules*
 As with relation rules = = true and = = false

If that sounds a little confusing, then have a look at the example program in figure 1. You see we built with our mixed up chemical bottles again! With only 5 targets and 4 conditions this is very simple and it is easy to see how the relations and implies sections are used.

It is always best to work through the targets in the order they are defined so that it also the order in which they will be tested. In this case we start flexible for a strong acid. For this to be the case, blue litmus must be turned strongly red. This is a rather subjective question, so the implies condition Blue Litmus Turned Red is considered first. In the relations section, we set 4 rules defined, the first saying that conditions 1 and 2 must be true and a second requiring condition 3 and 4 to be false. It is quite obvious in fact that if 1 and 2 are true then 3 and 4 will be false but this is defined in the implies section instead. The reason for doing it this way is that at some time the future it could be decided to resequence to test targets in a different order. This becomes difficult if the rules of the relations section are resequenced.

KBGEN program

The structure of this program is quite simple. The main program reads in the RAL source file line by line and looks for the comments in the expected order. Within each section the lines are validated according to the defined syntax. Any errors encountered are reported and if the error is fatal the processing is aborted. Standard references are used for skipping spaces and finding and extracting various elements in the source lines. It should not be that difficult to make changes to the program if desired.

The main obvious change to *KBGEN* is to open the output file on the same disk at the start of the program. As the format of the help, targets and conditions sections is very simple, these can be written directly to the output file after being validated so the large string arrays are avoided. This saving

[illegible]

DON'T LET THE ALSETH FILES GET TO YOU

FOOTMASTER IS THE ONLY POOLS PROGRAM AT HOME DOWN UNDER. AS WELL AS IN EUROPE. Every week and it guides you through the rough terrain of the Aussie leagues to successful predictions one and over again and again. It's a very successful connection from the ATRM ST program. Go on, don't hesitate, join the ATRM winners. It works for any game, cup or league either British or Australian or European. It has a massive database. It's not randomly, it gives you a choice at the best moments of the season. Perfect for week ahead or hour ahead. Join our members.

Do you believe in the power of the mind? If you answer "yes" to any of the above 3 questions then **POCIBUS** is for you. In **MAGIC PROJECTION** function will happily accommodate you. Do you believe in the power of the numbers? If the answer is "yes" then **POCIBUS** has something for you as well. In **WISDOMATHS BY TELEVISION** Powers will show you exact measurements. And for the rest of us mortals it will manipulate all the required information and it will give four different scenarios: a normal result forecast, a best, longer a best driver and a best, aware forecast. Are you into predictions, horoscopes and the great representatives? **POCIBUS** will give them to you, in fact it has everything that you need for a successful forecast. It is truly the ultimate and the state-of-the-art people program. So hurry order your copy today and see the people forecast.

DISCLOSURE: COSTS ONLY \$40!

SPECIAL OFFER: THE FIRST 24 ORDERS WILL RECEIVE A HOT GAME FREE OF CHARGE. DON'T MISS YOUR CHANCE TO GETTING A GOOD DEBATE. THE BOOKS ARE NOW ON SALE.

WANT CHICKEN? Pm PAYABLE TO "APOLONIA SOFTWARE" AND SEND THEM TO "APOLONIA SOFTWARE, UNIT 12, PARK
MOORE, 140 BATTERSEA PARK ROAD, LONDON SW8 5NL, UK" PLEASE STATE IF YOU WANT GIVE OR TAKE.

SALES BY THE BAY AREA FINE ART FRANCHISE. ALL OFFERS ARE SUBJECT TO STOCK AVAILABILITY.

ADDITIONAL SOFTWARE THE COMPANY THAT CARRIES

Dos 6.1

We put everything you have learned in this series to some practice use

By Fergal Moore

This final article of the series presents a complete disk operating system, which puts into practice some of the theory already learned. If you are interested, disassembly of the code from SC368 onwards should be useful.

Dos 6.1 - The Utility

This utility is in the same mould as DOS 5.1 on the domestic Commodore utility in America. This system belongs to a number of ways:

- 1) DOS 6.1 actually releases 175 new commands. Tables are now built using representing commands, and are used by ordinary users for all commands. This increases compatibility, space economy, and no need for a prefixing character.
- 2) The above feature means that DOS 6.1 commands can be used in program mode, unlike most normal commands.
- 3) DOS 6.1 provides some very powerful commands in an easily understood form.
- 4) You are left around 32,000 in the 4K block of memory from SC300-1000.
- 5) Loading is under ten seconds, providing you save out the code separately, and the program is easily copied to your own disk.

Users Guide

DISK ERROR STATUS

ERR displays the status of the disk drive at the current position on the screen. See your manual for a full explanation of error messages. Syntax: **ERR**

DOS6 COMMAND

DOS sends a command to the disk drive, opening up a range of around 30 disk commands. Again, see a manual for details of standard disk commands. If DOS is used without a

command, the error status will be returned.
Syntax: **DOS** "command"

DIRECTORY

Displays the directory of drive B or F (Usually drive B) without loading it into memory.
Syntax: **DIR**, drive number

MEMORY SAVE

Saves memory from start address to end address in the specified device. This is very useful with graphics or machine code data.
Syntax: **MSAVE**, start, end, "filename", dev

DEFAULT TO DISK

This sets drive B as the default device for all **SAVE** and **LOAD** operations. **LOAD**/**SAVE** will now have the same effect as **LOAD**/**SAVE** "B".
Syntax: **DISK**

DEFAULT TO TAPE

Types in now the default device for **SAVing** and **LOADing**.
Syntax: **TAPE**

NORMAL DEVICE NUMBERS

This returns to the normal Commodore device numbering system. Use this if you need to load with B.1.
Syntax: **NORMAL**

LIST BASIC PROGRAM

LIST allows you to list a Basic program directly off disk without loading it into memory. This is useful in checking a program, or grabbing lines from another program. **RUN/STOP** will stop the listing at any time. If you try to list a non-Basic program, you will get the expected package. Press **RUN/STOP** **KEYSTROKE** followed by **CLASSIC** to convert.
Syntax: **LIST** "filename"

LEAVE DOS 6.1

Leaves the DOS 6.1 system without disturbing Basic or any memory.

SYNOPSIS **SYNOPSIS** will return you to the DOS 5.1 system.

Syntax: **QUIT**

MEMORY USAGE

A memory map of the system could be useful.

C000 - C100 Program for adding new colors

C150 - C181 Table of jump addresses for new commands

C3000 - C33A New Dos routines
D33B - CFFF Free RAM for user programs

Some low memory and zero page locations are also altered during the command processing, but this has no effect on Basic and the area around the tape buffer is untouched.

Loading DOS 6.1

Type in and run the Basic program. When run, you will be presented with a screen of instructions. The machine code is being POKE'd into memory at this stage. On completion of the program, **NEW** itself and you are into the DOS system. Obviously, you will not want to keep leaving to load the basic loader each time you wish to use it. Therefore, I suggest you use the **MSAVE** command to save out the relevant two portions of memory.

USING DOS 6.1 IN YOUR OWN PROGRAM

As the commands can be used in your own programs it would be convenient to have to load DOS 6.1 in every case, especially if your programs are by others. Follow the following process first to include it in your program.

Load the two machine code files that you have saved at the start of your program. Immediately afterwards **SYNOPSIS** **SYNOPSIS** to initialize the commands.

Program Notes

When writing programs in the modified commandset, C64 Basic wants that you now have to enter colon after a **THEN** statement. There seems to be no way to override this, but it does not affect programs coded in the normal mode. These will work normally.

IF - IF GET A: IF A: - THEN GOTO B

I hope that this series of articles has been of use to you, and makes your job a little easier in the future.

200	4/27/2002	Cartier	EUR	1400	8	740	200P.1
	8/1/2004	27000					
201	4/27/2002	Cartier	EUR	1400	10	1070	1000P.1
	8/1/2004	27000					
202	4/27/2002	Cartier	EUR	1400	10	1070	1000P.2
	8/1/2004	27000					
203	4/27/2002	Cartier	EUR	1400	10	1070	1000P.3
	8/1/2004	27000					

Dos 6.1

```

10 40700 DATA 10 100 40 100 100 100
20 40710 DATA 100 0 40 10 100 100
30 40720 DATA 100 100 100 100 100 100
40 40730 DATA 100 100 100 100 100 100
50 40740 DATA 100 100 100 100 100 100
60 40750 DATA 100 100 100 100 100 100
70 40760 DATA 100 100 100 100 100 100
80 40770 DATA 100 100 100 100 100 100
90 40780 DATA 100 100 100 100 100 100
100 40790 DATA 100 100 100 100 100 100
110 40800 DATA 100 100 100 100 100 100
120 40810 DATA 100 100 100 100 100 100
130 40820 DATA 100 100 100 100 100 100
140 40830 DATA 100 100 100 100 100 100
150 40840 DATA 100 100 100 100 100 100
160 40850 DATA 100 100 100 100 100 100
170 40860 DATA 100 100 100 100 100 100
180 40870 DATA 100 100 100 100 100 100
190 40880 DATA 100 100 100 100 100 100
200 40890 DATA 100 100 100 100 100 100
210 40900 DATA 100 100 100 100 100 100
220 40910 DATA 100 100 100 100 100 100
230 40920 DATA 100 100 100 100 100 100
240 40930 DATA 100 100 100 100 100 100
250 40940 DATA 100 100 100 100 100 100
260 40950 DATA 100 100 100 100 100 100
270 40960 DATA 100 100 100 100 100 100
280 40970 DATA 100 100 100 100 100 100
290 40980 DATA 100 100 100 100 100 100
300 40990 DATA 100 100 100 100 100 100
310 41000 DATA 100 100 100 100 100 100
320 41010 DATA 100 100 100 100 100 100
330 41020 DATA 100 100 100 100 100 100
340 41030 DATA 100 100 100 100 100 100
350 41040 DATA 100 100 100 100 100 100
360 41050 DATA 100 100 100 100 100 100
370 41060 DATA 100 100 100 100 100 100
380 41070 DATA 100 100 100 100 100 100
390 41080 DATA 100 100 100 100 100 100
400 41090 DATA 100 100 100 100 100 100
410 41100 DATA 100 100 100 100 100 100
420 41110 DATA 100 100 100 100 100 100
430 41120 DATA 100 100 100 100 100 100
440 41130 DATA 100 100 100 100 100 100
450 41140 DATA 100 100 100 100 100 100
460 41150 DATA 100 100 100 100 100 100
470 41160 DATA 100 100 100 100 100 100
480 41170 DATA 100 100 100 100 100 100
490 41180 DATA 100 100 100 100 100 100
500 41190 DATA 100 100 100 100 100 100
510 41200 DATA 100 100 100 100 100 100
520 41210 DATA 100 100 100 100 100 100
530 41220 DATA 100 100 100 100 100 100
540 41230 DATA 100 100 100 100 100 100
550 41240 DATA 100 100 100 100 100 100
560 41250 DATA 100 100 100 100 100 100
570 41260 DATA 100 100 100 100 100 100
580 41270 DATA 100 100 100 100 100 100
590 41280 DATA 100 100 100 100 100 100
600 41290 DATA 100 100 100 100 100 100
610 41300 DATA 100 100 100 100 100 100
620 41310 DATA 100 100 100 100 100 100
630 41320 DATA 100 100 100 100 100 100
640 41330 DATA 100 100 100 100 100 100
650 41340 DATA 100 100 100 100 100 100
660 41350 DATA 100 100 100 100 100 100
670 41360 DATA 100 100 100 100 100 100
680 41370 DATA 100 100 100 100 100 100
690 41380 DATA 100 100 100 100 100 100
700 41390 DATA 100 100 100 100 100 100
710 41400 DATA 100 100 100 100 100 100
720 41410 DATA 100 100 100 100 100 100
730 41420 DATA 100 100 100 100 100 100
740 41430 DATA 100 100 100 100 100 100
750 41440 DATA 100 100 100 100 100 100
760 41450 DATA 100 100 100 100 100 100
770 41460 DATA 100 100 100 100 100 100
780 41470 DATA 100 100 100 100 100 100
790 41480 DATA 100 100 100 100 100 100
800 41490 DATA 100 100 100 100 100 100
810 41500 DATA 100 100 100 100 100 100
820 41510 DATA 100 100 100 100 100 100
830 41520 DATA 100 100 100 100 100 100
840 41530 DATA 100 100 100 100 100 100
850 41540 DATA 100 100 100 100 100 100
860 41550 DATA 100 100 100 100 100 100
870 41560 DATA 100 100 100 100 100 100
880 41570 DATA 100 100 100 100 100 100
890 41580 DATA 100 100 100 100 100 100
900 41590 DATA 100 100 100 100 100 100
910 41600 DATA 100 100 100 100 100 100
920 41610 DATA 100 100 100 100 100 100
930 41620 DATA 100 100 100 100 100 100
940 41630 DATA 100 100 100 100 100 100
950 41640 DATA 100 100 100 100 100 100
960 41650 DATA 100 100 100 100 100 100
970 41660 DATA 100 100 100 100 100 100
980 41670 DATA 100 100 100 100 100 100
990 41680 DATA 100 100 100 100 100 100
1000 41690 DATA 100 100 100 100 100 100

```


C O M P E T I T I O N

ENTER



No Strings Attached



Win wonderful prizes, care of Grandlam, in our easy-to-enter Thunderbirds Competition.

Five first prize winners - chosen after the closing date - will receive a Thunderbirds value and the game for Aamps or C34. Prizes and posters are available for ten

winners-up.

All you have to do is write the correct answers to the following questions and send them, on the back of a postcard or sealed envelope, to:

Thunderbirds Compo, Your Commodore, Argus House, Boundary Way, Bental Hempstead, HPM 1ST.

1. What is Bruce's real name?

- a) Hercules Hardbacker
- b) Bartoknew Baker
- c) Colbert Cline

2. What does F&B stand for?

- a) Fantastic Automobiles Birmingham
- b) Nothing
- c) Thunderbirds Air Ltd

3. What is Parker's surname?

- a) Popsy
- b) Shady
- c) Bender

Closing date, 31st August 1989

The Rules

Entries will not be accepted from employees of Arco Specialist Publications, Grandlam, or Taylor. The

competition also applies to employees families and agents of the companies.

The entry restrictions form part of the rules and the Editor's decision

is final. No correspondence will be entered into. In the event of a postal strike, we reserve the right to extend the closing date.

This Green and Pleasant Land!



Emma Norman dons her green outfit and, with banner grasped firmly in hand, campaigns for the organisation that everybody should know about

It's a novel advertising campaign the organisation Friends Of the Earth put five billion reasons why we should join them. The Human Race. There were five billion reasons could be given for joining the environmental organisation Greenpeace. Both Greenpeace and Friends of the Earth have the same object at heart.

To conserve the planet for future generations.

Most people have heard of Greenpeace and are aware of the work they do, but for those who do not know it is, in its own words 'An International Environmental Pressure group which maintains complete independence from all political parties anywhere.



in the world. Its aim is to bring an end to the pollution of the planet and to protect wildlife." It operates throughout the world.

It began in 1971 in Canada, with a protest voyage into a nuclear test zone at Amelika. The test was disrupted and the area is now a bird sanctuary. In 1977 Greenpeace opened its first branch in Britain where, to date, it has over 150,000 members. Today there are over 11½ million members worldwide with the majority of these coming from the USA. The Green is one of which they are very proud, but more members are always needed. Despite the successes, there are

many objectives still to be reached.

Why do we need Greenpeace? What is so harmful with dumping waste? The wastes are not confined to poisonous radioactive metals like Copper, Lead, Zinc Chlorine, Mercury and Arsenic. These build up in the environment and poison food sources. For example in Wood Spring, North Devon people were warned not to eat more than 1lb of Shrimps or 4oz of shellfish due to the toxins of cadmium (a harmful poison) therein. Two million tonnes per year of such waste is dumped in the Irish sea.

Although a direct connection has

not been established between waste-dumping and the virus that killed 17,000 seals last year (and appears to be re-emerging amongst the remaining Common Seal population), the possibility that pollution was the cause cannot be discounted and emphasises the urgency for action. There are countless examples of such atrocities caused by "Man's inhumanity to man", including Chernobyl, the Alaska oil spillage that killed the whales, elephants, the numerous toads, but what of the seagulls?

In the past few years Greenpeace has stopped boats of seal pups helped being about an and in commercial whaling, stopped the burning of hazardous waste in the North Sea, UK waters and the Mediterranean, stopped the dumping of radioactive waste at sea, helped persuade the British Government to spend millions of pounds cleaning the beaches in Britain and filtering the water from coal-fired power stations.

The majority of these achievements are brought about by their boaters. Greenpeace activists went out in dinghies and positioned themselves between the harpoons at the whaling boats and the fleeing whales. As a result, Commercial whaling is now banned in the North Atlantic. Greenpeace drew an inflatable dinghy under barrels of radioactive waste, therefore dumping of such waste at sea has now been stopped. Volunteers physically prevented baby seals being killed in the Orkney Isles and New Zealand by throwing themselves before the harpoon and the pups.

The level of the danger in which these people place themselves became apparent when one man was killed on the Rainbow Warrior when it was destroyed whilst trying to stop the testing of nuclear weapons off the coast of New Zealand. These people are risking their lives? What can we do to help?

You can join them in their fight to protect the planet by becoming a member, information and membership forms are available from Greenpeace 33-37 Juddville Green, London W1 6NR. (Telephone 01-235 4113 for a single, £17.50 for the family, £25 unshared, and £20 for overseas).

Greenpeace is at present campaigning for a nuclear free sea in Cornwall. Nuclear free Irish sea, stopping toxic waste dumping in Spain, working to protect the seals and dolphins round the British coast and trying to get Antarctica declared a World Park.

The Game



JOHN F. O'BRIEN, author of *Whales: The Natural History of the World's Largest Animals*, says a few points are so basic about the animal that it's almost silly. "Whales are mammals, not fish," he says. "They breathe air, have lungs, and give birth to live young. They are also warm-blooded and have a blubber layer under their skin. They are also the only animals that can hold their breath for so long." O'Brien is a professor of biology at the University of California, Santa Barbara.

The greatest danger to whales is the loss of their habitat. "Whales are highly dependent on the ocean for food, and the ocean is being polluted by human activities," says O'Brien. "Whales are also being hunted by humans, and this is a major threat to their survival." O'Brien is a professor of biology at the University of California, Santa Barbara.

Whales are also being hunted by humans, and this is a major threat to their survival. "Whales are also being hunted by humans, and this is a major threat to their survival," says O'Brien. "Whales are also being hunted by humans, and this is a major threat to their survival." O'Brien is a professor of biology at the University of California, Santa Barbara.

Whales are also being hunted by humans, and this is a major threat to their survival. "Whales are also being hunted by humans, and this is a major threat to their survival," says O'Brien. "Whales are also being hunted by humans, and this is a major threat to their survival." O'Brien is a professor of biology at the University of California, Santa Barbara.



RAINBOW WARRIOR

RAINBOW WARRIORS

The world's first environmental friendly software!

At last, an all action game that presents a solution to the environmental dilemmas faced by the entire human race. Rainbow Warriors is an action game with real depth and meaning. It simulates seven campaigns of Greenpeace members over the last few years.



*Mind
Benders*

GAMES FOR ADULTS

thus, share the same members and the rest.

The game features two different scenarios. You start saving two characters on each and perform various acts in order to liberate them, shoot bad guys. Thunderbush can never die and to catch the lead Anderson used and named as a hero. The first 'More Menace' involves Alan and James who have to rescue some trapped miners. This is not a simple operation as firstly has to shoot off the rock's passing station before it blows the whole mine, then rescuers will be rescued by a real train, causing earthquakes and chaos to seek other help. Like I said, Thunderbush cannot die.

The other three levels 'Sub Crash', 'Rail Job' and 'Countdown to Error' respectively, are all in a similar vein, although some races in which to complete your task get progressively more restrictive and the problems connected to the scenario become harder to figure out.

Graphics throughout are very good, and true to Anderson's original ideas. Presentation throughout is of a very high quality. For instance, before each mission there is a small digital animation sequence from the old episodes, for the four missions are all based on existing, single stories, although each, loosely, on those who won the Nobel



will find no longer in working. The famous three race: 3 4 1 2 1. Thunderbush, off Carl Ben, Ben, Ben Ben, via, has been rewritten to a rather good house version, which accommodates the rule sequence but leaves the game stuck in repeated sound effects.

Another small thing that I think helps every, my mistake about that game is the small animated sequences which the graphics go through at certain stages. From the Little Computer People giving the game their character. Each mission has a personal video tour, decorated in low passage straight into the system without Thunderbush's work.



work I think, Roger Grandlam and Gerry Anderson will all be very pleased with the final product of some of the best-looking and hard work. I know I am at a little thing Thunderbush has and I can see myself getting as happy by the computer screen as I did with the original series. Although it really is one of the oldest children in the book this time, for me, Thunderbush ARE good!

Rik Henderson

Progen are the programming team behind Chudra's Grade Teachers, and the more recent Progen proving that they are really an able adventure expert, although Thunderbush is the most complex of all, and probably the most complex.

Grandlam - £24.95



INFO

Computer: IBM
Graphics: VGA
Score: 95%
Availability: 40%
Original: 100%



GILBERT - ESCAPE FROM DRILL

Again Again - £9.99/£14.99

Follow Gilbert as he escapes in a spaceship as he explores the alien world while searching for the missing parts to the Multivision Dronium

THE case has another alien Gilbert, in his home once again for the bright lights of ITV. He has the contract, and the ego, was all he needs to do a local the missing parts of the Multivision Dronium and sign on the dotted line.

Gilbert has twenty-four hours to find the five missing parts, hidden by the discerning TV watchers of the planet, before some other (hopefully more talented, star signs) the contract. No Drilium possesses a full set of satellites and these would be viewers have actually left clues in various video games, which should help Gilbert fulfil his quest in satisfying the needs of the

Carbally young.

Gilbert starts his adventure in the middle of his home town, though why someone would wish to build a town in such a hostile place is beyond me. As you alight - you Gilbert has no feet - across town you are assaulted by all manner of beasts. Fortunately Gilbert can dispatch these creatures with a flask of the brew and a gladius of rage. The formidable method of defence is that he finds supplies of the sticky green stuff can be found in milk bars, along with the video games which give the missing parts. If Gilbert manages to shoot an entire wave of monsters, a flying

piece arrives, containing the missing parts. However, Gilbert must be able to fly, or swim, or crawl, or jump, or whatever, or he cannot use the missing parts. If he cannot, Gilbert must wait. So, making him so much the upper limit of the level and change levels when in the video system. During the rain brings Gilbert back to Drill with a heavy obviously made with the wrong kind of flour.

The two video games which Gilbert must complete offer a welcome

break from the more mundane pleasures of watching monsters in the video. Some of the better games include Earth Invaders, in which a one-eyed monster blasts at rows of descending humans.

Brain Drain - play against the computer to see who can watch the most ads. And Spotlight at the C&C Channel - watch how many failures to complete a video game produces a loss of one

point. (The video games are available on all video game systems.)

The missing parts are available in all video game systems, but a jungle, water and underwater - the best to occupy the mind. Seeking good has even the gameplay is a little repetitive.

Adrian Parnham

INFO

Category: 85%
Graphics: 82%
Sound: 85%
Usability: 88%
Overall: 79%

Eugen Varianova, leader of Mark Gornovitch and Richard Varianova, are the people responsible for the latest TV presenter computer game and as a publisher they have not all the best in a few weeks. Gilbert is there around what has been said.



SCRAMBLERS, in my opinion, would be so called because a couple of hours on the track of one of these things it's scrambled brains for you, dear. The only advantage posed here is one of less excitement as plot with even less pace than the real thing.

Different tracks of pure frustration lie before you, and failure to complete one within the time limit means a trip all the way back to the beginning. You start with the three dirt tracks beginning with the most serious track, A, which is designed to allow you to save time by letting you complete at least one of the three tracks. To complete the first track you simply have to ensure that the gear you are currently in (and of itself is correct for the slope you are travelling on) And that the bike you have the prettiest of controlling

does not exceed one of the two extremes of speed stalling, or going so fast you end up flying over the handlebars.

On the second dirt track you hit your first real obstacle, and find a use for the other controllable movement, the raising of the front and back wheels and at the same time I begin to add. This track also teaches you the position upon which the game is based, precise control over the bike's speed. When travelling over changes and rubble you must ensure that your speed does not exceed a narrow set of limits. Failure to do so results in the action freezing and a message informing you whether you went too fast or too slow. You may note that you do not even have the satisfaction of seeing your bike hit the dirt.

By the time you have reached the third track you must have mastered the raising and lowering of the wheels. If not you might as well turn your computer off and start again. Here the main obstacle comes in the form of small holes, which require you to raise your bike wheels whilst passing over them. Next come three tracks with beetles, the car variety, ships and wacky pits. On these courses the control over your speed becomes too crucial for your joystick to handle with any degree of accuracy and you inevitably crash.

If speed and your misfortune were all you had to worry about then life might even approach something like simple, but you also have to ensure that you keep on the course as indicated on a small strip of the screen which shows sideways movement.

If you are a masochist who enjoys the sound of revving engines then you might enjoy this game. But I doubt it.

Adrian Pampsey

Gremlin Graphics, one part of the *Gremlinware Software Systems*, are to return to *Super* Sheffield where, hopefully, more about will report.

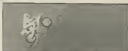
INFO

Gameplay: 80%
Graphics: 85%
Sound: 80%
Longevity: 45%
Overall: 54%

SUPER SCRAMBLE SIMULATOR

Gremlin - £9.99/£14.99

Mount up and put your wits against the latest game to bring you all the thrills and spills of high risk sport without the risks and without the thrills and even less of the sport





RED STORM RISING

Microprose - PC - £34.99

Tom Clancy's novel of World War III submarine warfare comes to life as you take the helm of a US Attack submarine. Sooner or later, it had to happen

IT'S BEEN more than 20 years since the Russian satellite, now called the USSR, was created. It has come to grips on the Atlantic coast with America. It is believed as the super powers divided would be inevitably divided by circumstance in the special North Atlantic. In an submarine warfare your strategy and skills will be vital in detecting and destroying enemy subs, task forces and troop carriers.

Earlier you take the such representation you can test your skills in a series of training missions, during which time you can learn how to use the ship's tactical layer in three display displays, the importance of keeping your speed low and a consistent depth to create good prospects doesn't reside in the water (and attack enemy subs), and how and when to use the weapon systems. The toughest part of submarine warfare is tracking down your enemy without being tracked. This means you can't use active sonar, which would give away your position. You must feel a great part of the ocean and listen for the signs that your enemy must use that. Once you have a good sonar contact (above 90% many torpedoes are fired at once

at past contacts) you can fire your torpedoes at a submarine, or missiles at the surface enemy.

Although you can play individual scenarios, the real challenge is when you play the Red Storm Rising Scenario and play your part in World War III. Here your missions are issued out and directly affect the handling of the war. Graphics sequences are the same as Navy battles against the Soviet onslaught. An example of this is when you're sent in to intercept a task force carrying troops. Not only do you have to track down and attack the convoy, your success in sinking the troop carriers will decide the success or failure of the Soviet attack. If you do badly, the map will indicate how bad.

As an idea, Red Storm Rising can't be faulted. It's based on a best selling book by a bestselling author and the computer implementation provides all the options and problems that face a modern submarine commander, who must do his best to win battles that reflect the outcome of the war. Unfortunately, the game falls down on presentation. Although the clips showing the progress of the war are good, the actual strategy

used in the battles (most of the game) are dull. In fact there's no comparison to games such as Silent Service or MS Attack sub. As a result this will deter many players from a challenging and interesting game.

Tony Hathornes

Red Storm Rising is an update on cooperation with Tom Clancy, by Red Alert who was the author of F-15 Strike Eagle and Silent Service. He was also the co-founder of Microprose.

INFO

CompuLink 84%
Graphix 12%
Sound 28%
Lemmings 49%
Grand 50%



STORMLORD

Hewson - £14.99 Disc, £9.99 tape.

AN evil Queen dominates the once beautiful and peaceful land. It was a land where fancies frolicked and bunch fauns lay around being bairns. It was also awfully dull, so it's just as well that the Queen arrived with her evil minions. Hadoosa creatures ruin the land, man-eating plants have taken root, and frolicking fauns lie captured in cages. Now Stormlord must rescue the fauns and liberate the land by destroying the Queen.

The land is represented on screen by a series of levels, each consisting of a number of sideways scrolling scenes.

Basic plants, static steps, and boulders form the platforms on which our hero can walk and jump. Control is somewhat limited although you can build up power for a bigger leap by holding down the fire button. The scenes also contain a very useful but unobtainable network of springboards that can hurl you into quite remarkable distances, and quick escape has by a chance in a lifetime. You can also be sure that there will be another springboard strategically placed for the return journey.

While Stormlord is running, jumping and leaping about the land various nasties are out to get him. Huge worms want to nibble him. Dragons want to fry him. Gnomes want him and Venus fly traps want to chomp him. However, he can get back with a well-timed, flying sword, but it's no substitute that it's best to get out of the way.

Fairy folk and foul creatures clash in a game that aims to take your screen by storm, as the hero Stormlord rescues rebellion against a malevolent monarch.

There are also a number of traps such as boulders and acid which drop down from above and doors which block your path throughout the game.

Luckily there are useful objects scattered around the landscape, but our hero and his incredibly quick hero can only carry one at a time, which I suppose isn't that surprising is it? You consider that a boy is almost as big as the door it opens! Apart from the keys,

there are other objects whose function is less obvious, which can make it difficult to always carry the one you need next, resulting in quite a bit of backtracking. This can be reduced by leaving them near springboards.

The result of all this is a game that's a cross between a sideways scrolling shoot-em-up and a platform game. Although I doubt this will appeal to other game's supporters as it is,

Freight man? Not only is that child pushing cart, but also out the side of their magic machine!



INFO

Copyright: 17%
Copyright: 40%
Points: 52%
Language: 38%
Overall: 41%

rather silly and equally unappealing.

Tony Hetherington

Stormlord was originally written by Raffaele Cirio (CSA conversion by Neil Jones) who also produced Cybermind, Cybermind II and Fusion.

ROCKSTAR ATE MY HAMSTER

WITH all the meat and soul of making it in the top Rockstar *Ate My Hamster* provides a gritty insight into this today-today world of rock 'n' roll management. Through the eyes of a somewhat shiny lump of no-talent

*Codemasters - £9.95 Tape
Sex, drugs and Rock 'n' Roll - one of these is featured in Codemaster's latest full price game. Does the game live up to its rather peculiar name?*

Gameplay: 88%
Graphics: 85%
Sound: 88%
Usability: 85%
Overall: 86%



intelligence, called Cecil Fox, you act as a manager looking and driving, and pulling the strings on the band's future plans. Not getting to the top is not so simple.

The ultimate goal of Rockstar, which has the intriguing looking of a board game, is to go Platinum by topping the charts, the world and elsewhere. With your Jewish subtrack Clive, who insists on using grocery talk, your first task is to choose the band. You are allowed up to four members and can select from a list of about 50 stars who appear as by now on a video screen to front of you.

It is a mistake that to choose the musicians of four band members as somewhere along the road to success, and the odd terrorist attack, tends to deplete the ranks. As for selecting stars, which are listed from a data cassette, it varies from player to player depending on who was beside the top gross stars and biggest budgets.

Back your band's album to some of the big names in the pop world including Britney Spears (Kylie Minogue), Diana (Hélie) and many more. With your band together it's time to buy the gear. New equipment costs money but it's one step to true professionalism, with amplifiers that go to eleven for that extra push over the d.f. However, being one better is not so easy as it

looks to them is the option of second-hand gear, at even lower is the dodgy knock off.

This is where your check to the top or lowly stroll into the wilderness begins. To start with you have four options, practice, gig, publicity or promo. It is probably advisable to fact them away for a bit of a gain, give them some publicity and then hit the road. The practice scenes cost money, so you have to choose your audience wisely. The publicity, which is obviously handled by the sceptical Clive, involves the aptly named *Don't Stoppage* and includes a public speaking of 1000 addresses, including a certain 1000 during a tour. Publicity costs, however, do not always get covered by the press, but that's a chance you have to take.

If you decide to go you have to select a venue from the list of pubs, clubs, universities, concert halls and stadiums. For the first time out it is probably best to play to cheap and easy, in-100 venues. A decision on the ticket price has to be reached and the number of tickets played. After that, Clive informs you of the number of tickets that were sold and consequently the amount of money made. At the start of the game money is the key to success, especially if you have chosen band members with high salaries, they have to

be paid on a weekly basis.

If you are going well and you have succeeded in front page gutter news, the phone will start to ring with offers of record contracts, chart, gigs and potential sponsorships. When a satisfactory record deal is reached you have the option of selling the vinyl and shooting a promo to begin your climb on the record sales market and a position in the national charts. The prospect of shooting a video means more money as producers, locations and stage details with every band's budget.

Success is important for the band's future. With some surprisingly recognizable names the band under your guidance will enter the nation of the weekly charts and face the problem of staying at the top, paying taxes and caring for the fans.

Rockstar *Ate My Hamster* is an enjoyable game even though the graphics are not particularly good. It is addictive with reasonable good value for its rubber and price, and for the sheer fun of the music world.

Mark Jones



Celia Jones, the programmer of all the formats of Rockstar, is undoubtedly and without doubt in one way. With

ULTIMA TRILOGY

*Origin Systems (Microprose) - PC £29.95.
C64 £24.95 disk only.*

*More than one million global gamers
have delved into the Ultima series of role
playing games. Now the first three are
available in a special edition.*

THE Ultima Trilogy is a special edition of the infamous Ultima games in which you must confront a Triad of Evil who threaten the land ruled by Lord British. Both Ultima I and II have been available in the UK before but this is the debut for Ultima III which will provide every state lover of role playing whilst filling some of the gaps and answering some questions that were left open with its ancestors.

The content of the box is impressive and immediately gives the impression that this is a game to play. Alongside the game disks (1 for C64, 4 for PC) there is a quarterest guide, a player reference guide that outlines the keyboard controls and available spells, and a massive 186 page instruction manual that covers everything from the back of the amber scales to a guide to the mountains and oceans you will face. It also contains three full colour maps of the lands you must save.

In the games you play a brave adventurer who may be of human, elf, dwarf or hobbit stock, and be named to the art of a fighter, cleric, thief or wizard. Throughout the games you have a controlled through a series of single key presses that correspond to commands. For example, G lets you offer gold as a bribe, U unlocks doors if you have the key, and I splits a

torch. Although you begin alone with a number of points to distribute between your role playing characteristics, you will soon find others that will join you in your quest.

Ultima I is the first age of darkness and features a desperate battle against hordes of nightmarish creatures spawned by the Wizard Marston. The delusional people of Sosaria fix on you as the newest strength and all will be lost if a hero is not found.

The same heroic mission will be put to the test in Ultima II when Minax, the young apprentice of Marston, seeks to avenge her father's death. Such is the power of this Enchanter that her efforts to find her father's slayer have seen rifts in the fabric of time. This has opened doors through which a brave adventurer might bring about her doom.

In Ultima III the third part of the Triad of evil is revealed, as fragments of a manuscript hint at an unlikely alliance between Marston and Minax and of a demon-games that there since had produced. This is the ultimate test for a hero as not openly under the land and your foe is unknown apart from the power it possesses.

The games look a little dated now compared to the more recent Ultima IV and V, and other games such as Pool of Radiance and



Lord's Tale III, but this special edition is a must for Ultima players looking to complete the story.

Tony Hottenstages
The Ultima adventures
were written by Richard
Garriott between 1979 and
1983 from being published



through Gamma-on-Line and then through Origin, the company he co-founded. The first Ultima game to appear in the UK was imported by US Gold (Ultima III) since then others have been imported by Microprose.



Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session. Listening one of the superb programs from *Four Commodore*. Your fingers reach for the keyboard and press the letters H, U and N. You press RETURN, at last and nothing happens.

Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little but sleep through unnoticed.

The *Four Commodore Software Service* also available all of the programs from disk now on both cassette and disk at a price of \$4.00 for disk and \$4.00 for cassette. None of the documentation for the program is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then hard copies are available from the following address:

INFONET LTD, 5 River Park Drive, Beckenham, Kent,
BPM 1HL
Tel: (0447) 35661

Please contact this address for prices and availability

The Disk

Programs on the disk will also be supplied as totally working versions, i.e. when possible we will not use Basic Loaders that make use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 cassette ROMs; programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strip containing the article type: C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strip. The symbols have the following meaning:



This symbol means that the program is available on cassette.



These programs are available on disk

Please Note

Since the programs supplied on cassette are total working versions of the programs, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

APRIL 1989

BASEX - Give your C64 new sound, graphics and tacit commands as well as a machine code assembler with the Basic reference.

AUTOSCHOOL - Professional text screen scrolling with the C64 utility.

BALANCE SHEET - Keep your bank manager happy by keeping better track of your money with this C64 program (disk only).

BORN 64 - Add more control to your own programs with the C64 utility.

88 COLUMNS - A suite of Basic graph commands for C128 88-column users.

EFUS - A disk file copier for Plus/4 users. Available on disk only.

MULTI-PRECISION ARITHMETIC - A suite of powerful mathematical routines expanding your C64's number cranking abilities.

STORAGE SPACE - Use the RAM behind the C64's External and Basic ROMs for data storage (available on disk only).

ORDER CODE

DISK YCAPR89 04 00

TAPE YCAPR89 04 00

MAY 1989

ANTI-FREEZE - Protect your C64 programs from cartridge head freezes.

MULTI-COLOURED LIST - Brighten up your C64 Basic listings by adding convenient colour.

PRINTX - A suite of powerful screen plotting commands to add to your Plus/4's Basic (disk only).

RANKER 118 Money Management for C128 disk users
TURNED TOUCH - A superb typing trainer for C64 disk users

ORDER CODE
DISK TDMAY89 £6.00
TAPE TCMAY89 £4.00

JUNE 1989

RETURNER - A handy routine that allows your C64 programs to reload a disk-based menu (disk only)
LINE INPUT - Improves your C64's INPUT routine
PROGRAM ANALYSER - A range of programs to help you examine your C64 Basic programs in depth
ASM ASSEMBLER - A powerful £100 assembler for the C64

ORDER CODE
DISK TDMJUN89 £4.00
TAPE YCJUN89 £4.00

JULY 1989

MOUSE 48 Use a 1931 mouse in 80 columns mode on the C128
QUICK SEARCH - Add a FIND option to C64 Basic
SPRITE LIBRARY - Add a collection of sprites to your growing sprite library (C64)

HEAD FOR HOME - An update on our popular C16 and Plus/4 game. Available on disk only
WINDOWS ON THE 64 - Powerful windowing commands for the C64
SPRITE SPRINT - Give your Plus/4 or expanded C64 sprite - just like a C64. Available on disk only

ORDER CODE
DISK YCJUL89 £4.00
TAPE YCJUL89 £4.00

AUGUST 1989

REASONING ON THE 128 - Programs for the first part of our expert system series (128 Disk only)
THE SAURUS - Build up your own C64 thesaurus with this program for disk or tape
WINDOWS ON A NAME - Generating names on your C64
ADDRESS BOOK - A memory-based address book for C64 users
DISK SCRAMBLER - Protect your disk data from prying eyes (C64 disk)

ORDER CODE
DISK YCAUG89 £6.00
TAPE YCAUG89 £4.00

Coverlets or disks are available from March 1989. Please ring the editorial office (01-407 0626) for details of these.

ORDER FORM - PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
SEPT 1989		TAPE (£4.00)	YCSEP89	
SEPT 1989		DISK (£4.00)	YDSEP89	
OVERSEAS POST £1				
			TOTAL	

NAME

ADDRESS

POSTCODE

I enclose a cheque/postal order for £... made payable to ARGUS SPECIALIST PUBLICATIONS LTD

All orders should be sent to: FOUR COMMODORE

READERS SERVICES, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS HP3 3ST

Please allow 28 days for delivery.

TERRAMEX

Grand Slam - \$2.99 Tape

High in the icy vastness of space, something is headed our way. A giant asteroid is on collision course with Earth and only you can save the world.

WELL, actually the only person who can save the world is Professor Eysarua, who produced the asteroid's coming some 30 years ago. The trouble was that no one believed him, so he went off to a dark and dingy corner to telt.

Taking the part of one of five stereotyped explorers, you must search out the laser of Lamon, and then search out Professor Eysarua so he can build an asteroid deflector. Although if the world is saved then Grand Slam will be able to write more programs like this. . . Well, total destruction might not be so bad as it's often said.

When Terramex first appeared on the ST and Amiga it had the feel of an average 8-bit game, and now we see it as budget 8-bit fare, which is better value for money. It doesn't make the game any more interesting though.

Having selected your character and wound for the rest of the game to lead you on off into the dark unknown lands and by your sensible means, an unimpressive 2-D arcade adventure style. An irritating poor of music will leave your teeth grinding by now, so go for the volume switch whilst concentrating on the available but colorful characters, and the generally dull

landscapes.

One of the first objects encountered, and the game's object uncoverer, is a vacuum cleaner, which endows the power of flight. Yes, you'll believe it however can fly, but not everywhere. One of the features of Terramex then becomes obvious: you can hover along a certain distance and suddenly because it would make the game too easy, you stop in mid air, unable to progress further. There isn't actually anything there stopping you from going on, you just can't. One of the neat objects needed is an umbrella which acts as a parachute. One good point is that you can ask your character for an idea when faced with a problem. If you're lucky the right object will flash up, and then it's down to you to use it properly.

Besides the object manipulation part, there's also the challenge of having to leap over snakes, cactuses, and other obstructions, as an lady Jones like woman. Should all efforts fail, then a quite humorous sequence follows whereby the asteroid plunges into the Earth. For these incidents this takes some testing.

For aficionados of the arcade adventure, £2.99 represents good value, if spectacular entertainment, but for those not converted to the cause there is nothing here to convince that an expedition to find Professor Eysarua is a better idea than waiting at a bar for character to fall.

Duncan Evans

INFO

Golfing: 68%
Crusader: 72%
Swords: 67%
Legislative: 71%
Olympic: 69%

64

Grand Slam started life quite simply, with Terramex being one of its poorer schemes. Arcos offerings are generally much better.



TIME Lord has always been round his, out of the time remastered map. At the time of the great cascade, the Fire Wizard happened to be away, but news soon spread through the town that he has been horribly murdered. The Earth Mage has died, the Water Mage has disappeared and the Wind Mage has been previously wounded. Rumours of an enormous magical beast abound as food supplies start to run danger with low. In desperation, the mayor sends forth a hero who promptly disappears. Now you have been awakened to help.

You have the choice to play one of six characters, each differing in three strengths: armour and magical abilities. These attributes can be able to temporarily with the help of magic, or permanently by

finding relics — something that you will have to do if you are to get very far in the game.

Fire King follows the usual role playing ideas in so much as you have to master basic fighting, manage and finding key objects. There are books scattered around containing clues. Although some information, especially in the Tibbers book, has to be paid for.

The key to the game lies in manipulating the objects that you find and so purchase. You have seven pockets each of which can contain up to four similar objects. It's like death spells etc. The problem is that there are usually more than one type of objects to carry, so you will need to plan ahead. On top of that, there is food to buy which will help to top up any energy, but is fairly



FIRE KING

Electronic Arts, Price £18.99 Disc
Role-playing done under is the name of the game. Gordon Hamlett casts a clairvoyance spell and determines if everything is for dunkum or not.



expensive. In the game books and plays a like. I wonder if you can produce more of that, there is more of a feeling of adventure in the game rather than just one large discovery.

The problem with Fire King is the manipulation of objects. The pocket system is very cumbersome in use especially in a fast pace situation. The designers are very clever and I suspect that fans of role playing games will find this too much of an arcade game for their liking. I must confess that knowing their previous track record I expected something a bit more original from this company.

Gordon Hamlett

INFO	
Concept	40%
Graphics	50%
Sound	40%
Usability	50%
Value	50%

Written by our *Gameplay* Column called *More Fun and Excitement through 320* — there are many more...

Extending Basic

Get a printout of the variables of your Basic program with this latest routine in our series on extended Basic

By Burghard-Henry Lehmann

In this series of articles on developing extended Basic routines, we have already dealt, several times, with the way Basic stores its variables. Therefore, the next logical step is to write a routine which gives us a printout of the variables created by a Basic program and what they contain at any given time.

This is a very easy thing to do, because all that's required is that we loop through the Basic variable area and print each variable name and content.

I wanted to make this facility possible as a direct command and in connection with the trace facility which we developed in the last article. I am including this in this installment for all you people who don't buy last month's **YOUR COMMODORE** (please, no post!)

As a direct command, you simply

enter **VARL**, and all variables of the Basic program which you have just run are printed out. Of course, if you haven't run a Basic program or have given the direct **CLR** command, which clears all Basic variables, nothing will be printed out. The same is true if you have changed the Basic textfile. As you know, the moment you enter a Basic line by pressing return all variables are discarded.

In connection with the trace facility which we developed last month, you can call a printout of all variables after each line has been executed and the trace routine has regenerated the line at the top of the screen. You will remember, that after each listing step, the computer waits until you press any key. To get a printout of the variables simply press "V". If you want program execution to continue press any other key.

How Basic Stores Its Variables

Before we look closer at the program itself, let's briefly reorganize how Basic stores its variables.

All Basic variables are stored directly after the end of the Basic textfile. To find the start of the variable area you have to peek the next page variable decimal 41416. Hex 10176. Each variable takes up seven bytes of memory.

The first two bytes contain the variable name. The way the variable name is stored shows what type of variable we are dealing with.

A floating point variable has both bytes of its name entered as single ASCII. If the name consists of only one letter, the second byte contains zero.

An integer variable has 128 added

is both its name character, and if there is only one character (which, as you remember, always has to be a letter), the second byte contains 128. (All this means, that in each case, the area of the eight binary bits is set.)

A string variable is distinguished by having the letter of its name entered in simple ASCII, while the second character has 128 added to the ASCII value, or 128, if there isn't a second character.

Finally, a function definition has 128 added to the first letter of its name, while the second character is simple ASCII or zero. (Incidentally, in the program given I have ignored function definitions. As always, that is partly due to my laziness and partly in order to encourage you to add this to the routine yourself.)

The final five bytes of each variable contain the contents of the variable itself. This is entered in the area of the string variable. Here, the third byte contains the length of the string, and the fourth and the fifth bytes contain the base address of the string in the Basic textfile.

This is quite a clever way of going about things. Instead of moving the string itself in the variable area, which would of course take up a lot of memory and make the whole variable area pretty difficult to manage, a pointer address which takes up only two bytes is stored in the variable area. The disadvantage with this method is of course, that the Basic textfile must not be changed. The moment it is changed, the pointer to a certain string in the variable area might not be correct any more. This is one of the reasons why you can't continue running a Basic program after you have changed the textfile.

The storage of an integer variable is also pretty straightforward. The value of the variable is stored in the third and fourth byte after the name, whereby the third byte contains the high byte of the value, plus the so-called sign bit, and fourth byte contains the low byte. You assign an integer variable by entering '%' after a, e.g. N%.
An integer ranges from -32768 to +32767. If you add those two numbers up you get 65535, which is 65536 short of one, because the zero is included as a valid number. And 65536 is the

maximum range of a 16-bit address. If it is seven, or the sign bit, of the high byte is set, the number is considered to be negative, if it is clear the number is considered to be positive. For example, high 0 and low 100 represent +100, while high 255 and low 128 represent -100.

The most complicated variables of all are the floating point variables. This is a pretty big subject which I do not want to enter at this point. There is also no need to, because in our program we are using ROM-routines to convert the floating point number as it is stored in the variable area into an ASCII string which we then simply print out.

But, for the mathematicians amongst you, here is how a floating point number is stored in the Basic variable area. The byte after the variable name contains the exponent, the byte after that contains the first mantissa, plus the sign bit, and the final three bytes contain mantissa two to four.

Printing the Variables

The routine which prints each variable name and the contents of the variable, is a straightforward, that is, a very little explanation.

The main loop, which I call VARSLCOP, first of all prints the variable name and decides at the same time what type of variable has to be dealt with (lines 4008-4010). According to these tests the program flow then jumps to the appropriate routine: PPRINT, if it is a floating point variable, INTPRINT, if it is an integer variable, and STRING, if it is a string variable.

Before the value of each type of variable is printed, some spaces and an equal sign are printed, as well as the 'Type' in case of a string, and the '%-sign' in case of an integer. This makes the printing most presentable.

In the case of a floating point variable, A is printed at the low byte of the first byte of the floating point value and Y is printed at the high byte (lines 4070-4074).

Then the ROM-routine at 1698A2 is called to put the value into the first of the two floating point accumulators.

Next, the ROM-routine at 16DD0 outputs the contents of the

floating point accumulator into a previous ASCII string. On exit from this routine, A and Y point to the base of that string.

Finally, the string is printed in the usual fashion, with PRINT, which prints any string point at by A/Y and terminated with a zero.

An integer variable is printed by loading its high byte into A and its low byte into X. The ROM-routine at 16A1D0, which prints any number contained in A/X, prints the integer value out.

To print a string variable, we first get the length of the string and store it in Z% (line 3490-3492). Then we transfer the address which points at the string in the textfile into area page 215/214 (lines 3510-3512).

Finally, we use routine-Y to print the string, letter by letter. This is done with a loop which is limited by the length value contained in Z% (lines 3600-3602).

After a variable has been dealt with, VARSLCOP ends by incrementing the pointer address by seven (lines 4076-4078). The vital loop can test, which tells when there are no more variables, is done at the top of VARSLCOP (lines 4078-4100). This is because the routine has to be called as soon as there are no variables left.

The end of the Basic variable area is contained in area page 51P/50. If this point is reached, there are no more variables. What follows are the Basic arrays with which we are not dealing at this point.

Final Notes

As I was developing the variables printing routine I discovered in my great excitement that certain variable names did not work properly. For example 'CC' was stored as 'C' and 'CI' didn't work at all and resulted in a syntax error report.

Then I discovered the reason for these complications. The extended Basic commands we can start with a letter, that is a normal ASCII character. If you declare a variable by entering the LET command (or can usually check), then the variable name has to start with a letter!

To overcome this I changed the main routine in the following way. First, I added two lines, which set for

PROGRAMMING

2700	END END	2711	END PROGRAM	2720	END
2705	END PROGRAM	2712	END	2725	END
2710	END	2713	END	2730	END
2715	END	2714	END	2735	END
2720	END	2715	END	2740	END
2725	END	2716	END	2745	END
2730	END	2717	END	2750	END
2735	END	2718	END	2755	END
2740	END	2719	END	2760	END
2745	END	2720	END	2765	END
2750	END	2721	END	2770	END
2755	END	2722	END	2775	END
2760	END	2723	END	2780	END
2765	END	2724	END	2785	END
2770	END	2725	END	2790	END
2775	END	2726	END	2795	END
2780	END	2727	END	2800	END
2785	END	2728	END	2805	END
2790	END	2729	END	2810	END
2795	END	2730	END	2815	END
2800	END	2731	END	2820	END
2805	END	2732	END	2825	END
2810	END	2733	END	2830	END
2815	END	2734	END	2835	END
2820	END	2735	END	2840	END
2825	END	2736	END	2845	END
2830	END	2737	END	2850	END
2835	END	2738	END	2855	END
2840	END	2739	END	2860	END
2845	END	2740	END	2865	END
2850	END	2741	END	2870	END
2855	END	2742	END	2875	END
2860	END	2743	END	2880	END
2865	END	2744	END	2885	END
2870	END	2745	END	2890	END
2875	END	2746	END	2895	END
2880	END	2747	END	2900	END
2885	END	2748	END	2905	END
2890	END	2749	END	2910	END
2895	END	2750	END	2915	END
2900	END	2751	END	2920	END
2905	END	2752	END	2925	END
2910	END	2753	END	2930	END
2915	END	2754	END	2935	END
2920	END	2755	END	2940	END
2925	END	2756	END	2945	END
2930	END	2757	END	2950	END
2935	END	2758	END	2955	END
2940	END	2759	END	2960	END
2945	END	2760	END	2965	END
2950	END	2761	END	2970	END
2955	END	2762	END	2975	END
2960	END	2763	END	2980	END
2965	END	2764	END	2985	END
2970	END	2765	END	2990	END
2975	END	2766	END	2995	END
2980	END	2767	END	3000	END
2985	END	2768	END	3005	END
2990	END	2769	END	3010	END
2995	END	2770	END	3015	END
3000	END	2771	END	3020	END
3005	END	2772	END	3025	END
3010	END	2773	END	3030	END
3015	END	2774	END	3035	END
3020	END	2775	END	3040	END
3025	END	2776	END	3045	END
3030	END	2777	END	3050	END
3035	END	2778	END	3055	END
3040	END	2779	END	3060	END
3045	END	2780	END	3065	END
3050	END	2781	END	3070	END
3055	END	2782	END	3075	END
3060	END	2783	END	3080	END
3065	END	2784	END	3085	END
3070	END	2785	END	3090	END
3075	END	2786	END	3095	END
3080	END	2787	END	3100	END
3085	END	2788	END	3105	END
3090	END	2789	END	3110	END
3095	END	2790	END	3115	END
3100	END	2791	END	3120	END
3105	END	2792	END	3125	END
3110	END	2793	END	3130	END
3115	END	2794	END	3135	END
3120	END	2795	END	3140	END
3125	END	2796	END	3145	END
3130	END	2797	END	3150	END
3135	END	2798	END	3155	END
3140	END	2799	END	3160	END
3145	END	2800	END	3165	END
3150	END	2801	END	3170	END
3155	END	2802	END	3175	END
3160	END	2803	END	3180	END
3165	END	2804	END	3185	END
3170	END	2805	END	3190	END
3175	END	2806	END	3195	END
3180	END	2807	END	3200	END
3185	END	2808	END	3205	END
3190	END	2809	END	3210	END
3195	END	2810	END	3215	END
3200	END	2811	END	3220	END
3205	END	2812	END	3225	END
3210	END	2813	END	3230	END
3215	END	2814	END	3235	END
3220	END	2815	END	3240	END
3225	END	2816	END	3245	END
3230	END	2817	END	3250	END
3235	END	2818	END	3255	END
3240	END	2819	END	3260	END
3245	END	2820	END	3265	END
3250	END	2821	END	3270	END
3255	END	2822	END	3275	END
3260	END	2823	END	3280	END
3265	END	2824	END	3285	END
3270	END	2825	END	3290	END
3275	END	2826	END	3295	END
3280	END	2827	END	3300	END
3285	END	2828	END	3305	END
3290	END	2829	END	3310	END
3295	END	2830	END	3315	END
3300	END	2831	END	3320	END
3305	END	2832	END	3325	END
3310	END	2833	END	3330	END
3315	END	2834	END	3335	END
3320	END	2835	END	3340	END
3325	END	2836	END	3345	END
3330	END	2837	END	3350	END
3335	END	2838	END	3355	END
3340	END	2839	END	3360	END
3345	END	2840	END	3365	END
3350	END	2841	END	3370	END
3355	END	2842	END	3375	END
3360	END	2843	END	3380	END
3365	END	2844	END	3385	END
3370	END	2845	END	3390	END
3375	END	2846	END	3395	END
3380	END	2847	END	3400	END
3385	END	2848	END	3405	END
3390	END	2849	END	3410	END
3395	END	2850	END	3415	END
3400	END	2851	END	3420	END
3405	END	2852	END	3425	END
3410	END	2853	END	3430	END
3415	END	2854	END	3435	END
3420	END	2855	END	3440	END
3425	END	2856	END	3445	END
3430	END	2857	END	3450	END
3435	END	2858	END	3455	END
3440	END	2859	END	3460	END
3445	END	2860	END	3465	END
3450	END	2861	END	3470	END
3455	END	2862	END	3475	END
3460	END	2863	END	3480	END
3465	END	2864	END	3485	END
3470	END	2865	END	3490	END
3475	END	2866	END	3495	END
3480	END	2867	END	3500	END
3485	END	2868	END	3505	END
3490	END	2869	END	3510	END
3495	END	2870	END	3515	END
3500	END	2871	END	3520	END
3505	END	2872	END	3525	END
3510	END	2873	END	3530	END
3515	END	2874	END	3535	END
3520	END	2875	END	3540	END
3525	END	2876	END	3545	END
3530	END	2877	END	3550	END
3535	END	2878	END	3555	END
3540	END	2879	END	3560	END
3545	END	2880	END	3565	END
3550	END	2881	END	3570	END
3555	END	2882	END	3575	END
3560	END	2883	END	3580	END
3565	END	2884	END	3585	END
3570	END	2885	END	3590	END
3575	END	2886	END	3595	END
3580	END	2887	END	3600	END
3585	END	2888	END	3605	END
3590	END	2889	END	3610	END
3595	END	2890	END	3615	END
3600	END	2891	END	3620	END
3605	END	2892	END	3625	END
3610	END	2893	END	3630	END
3615	END	2894	END	3635	END
3620	END	2895	END	3640	END
3625	END	2896	END	3645	END
3630	END	2897	END	3650	END
3635	END	2898	END	3655	END
3640	END	2899	END	3660	END
3645	END	2900	END	3665	END
3650	END	2901	END	3670	END
3655	END	2902	END	3675	END
3660	END	2903	END	3680	END
3665	END	2904	END	3685	END
3670	END	2905	END	3690	END
3675	END	2906	END	3695	END
3680	END	2907	END	3700	END
3685	END	2908	END	3705	END
3690	END	2909	END	3710	END
3695	END	2910	END	3715	END
3700	END	2911	END	3720	END
3705	END	2912	END	3725	END
3710	END	2913	END	3730	END
3715	END	2914	END	3735	END
3720	END	2915	END	3740	END
3725	END	2916	END	3745	END
3730	END	2917	END	3750	END
3735	END	2918	END	3755	END
3740	END	2919	END	3760	END
3745	END	2920	END	3765	END
3750	END	2921	END	3770	END
3755	END	2922	END	3775	END
3760	END	2923	END	3780	END
3765	END	2924	END	3785	END
3770	END	2925	END	3790	END
3775	END	2926	END	3795	END
3780	END	2927	END	3800	END
3785	END	2928	END	3805	END
3790	END	2929	END	3810	END
3795	END	2930	END	3815	END
3800	END	2931	END	3820	END
3805	END	2932	END	3825	END
3810	END	2933	END	3830	END
3815	END	2934	END	3835	END
3820	END	2935	END	3840	END
3825	END	2936	END	3845	END
3830	END	2937	END	3850	END
3835	END	2938	END	3855	END
3840	END	2939	END	3860	END
3845	END	2940	END	3865	END
3850	END	2941	END	3870	END
3855	END	2942	END	3875	END
3860	END	2943	END	3880	END
3865	END	2944	END	3885	END
3870	END	2945	END	3890	END
3875	END	2946	END	3895	END
3880	END	2947	END	3900	END
3885	END	2948	END	3905	END
3890	END	2949	END	3910	END
3895	END	2950	END	3915	END
3900	END	2951	END	3920	END
3905	END	2			

YOUR COMMODORE

Advertiser: 0442 66651
(1-VA1)

Some days only, 01835 plus VAT per ad line in column
comprehensive electronic form. Ring for information on
rates and charges for inserts.

All advertisements in this section should be placed in
an advertisement is accepted, subject to the terms and conditions
printed on the advertisement rate card (available on request).



0442 66651

Send your requirements to
CLASSIFIED DEPARTMENT
ASP LTD, ARGUS HOUSE,
BOUNDARY WAY,
Hemel Hempstead HP3 7ST

SOFTWARE

ONLY POOLS AND HORSES

FOOTBALL COMMODORE (1416) £42.00 (1 year)

A FOOTBALL COMMODORE is now the most popular football software
available for the Commodore 64. It is a complete football simulation
program which can be played on any Commodore 64 system.
It is a complete football simulation program which can be played on any
Commodore 64 system. It is a complete football simulation program
which can be played on any Commodore 64 system.

BIODIVERSITY - HANDICAP YOUNG (1416) £42.00 (1 year)

This program is for the young. It is a complete biodiversity simulation
program which can be played on any Commodore 64 system. It is a
complete biodiversity simulation program which can be played on any
Commodore 64 system. It is a complete biodiversity simulation
program which can be played on any Commodore 64 system.

PRECISE TONALITY (1416) £42.00 (1 year)

A complete TONALITY program is now the most popular tonality
simulation program available for the Commodore 64. It is a complete
tonality simulation program which can be played on any Commodore
64 system. It is a complete tonality simulation program which can be
played on any Commodore 64 system.

Write now to: **ASP LTD, Argus House, Boundary Way, Hemel Hempstead, Herts HP3 7ST**

0442 66651 (1-VA1) (1-VA1) (1-VA1) (1-VA1) (1-VA1) (1-VA1) (1-VA1) (1-VA1) (1-VA1) (1-VA1)

Write now to: **ASP LTD, Argus House, Boundary Way, Hemel Hempstead, Herts HP3 7ST**

HARDWARE

GOO LTD. GEM PRINTER

BARGAINS & OCP ART STUDIO



GOO LTD. GEM PRINTER

BARGAINS & OCP ART STUDIO

GOO LTD. GEM PRINTER

BARGAINS & OCP ART STUDIO

GOO LTD. GEM PRINTER

BARGAINS & OCP ART STUDIO

GOO LTD. GEM PRINTER

BARGAINS & OCP ART STUDIO

GOO LTD. GEM PRINTER

BARGAINS & OCP ART STUDIO

SERVICES

**FOR DETAILS ON
ADVERTISEMENT
RATES
CALL
0442 66651**

SPECIAL OFFERS

64TIN USERS

64TIN USERS: 64TIN is a new software
package which can be used on any Commodore 64 system.
It is a complete 64TIN simulation program which can be played on any
Commodore 64 system. It is a complete 64TIN simulation program
which can be played on any Commodore 64 system.

ADVERTISERS

**TO
ADVERTISE
IN
YOUR
COMMODORE
CALL
0442 66651**

YOUR COMMODORE SPECIALS

TECH DRAW 64 - a
comprehensive technical
drawing software for the Commodore 64.
It is a complete technical drawing software
program which can be played on any Commodore 64 system.
It is a complete technical drawing software program
which can be played on any Commodore 64 system.

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

TECH DRAW 64 - a

COMPUTER GAMES

The complete list of all the best
computer games available for the Commodore 64.
It is a complete list of all the best computer games
available for the Commodore 64. It is a complete list of all the
best computer games available for the Commodore 64.

PUBLIC DOMAIN SOFTWARE

Public Domain Software: Public Domain Software is a new
software package which can be played on any Commodore 64 system.
It is a complete Public Domain Software program which can be played on any
Commodore 64 system. It is a complete Public Domain Software program
which can be played on any Commodore 64 system.



**BARDALE
COMPUTERS LTD**

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

BARDALE COMPUTERS LTD

THE *Commodore* is not just an elite machine for the rest of the market. **THE** *Commodore* is also a machine that can do more. The new *Commodore* can do it all. It can read all the great books that are floating around the *Commodore* market.

The Lines are Blurred

A Reading resident has been having a rather harrowing time of late due to an American magazine who misprinted the Activision hotline number. The givings in question was receiving calls all around the clock from people requesting hints and tips on Activision games among some distress. The correct number is 877.944.500000.

Squalling at telephone lines, those kind puns at British Telecom have changed Initiative's number. The new number and code is, if you're at all interested, (07342) 511780.

Arthur S. Lee

Jim Ford, the editor of *Electronics Today International* - one of our many solid customers, from the Home Bu-

to the recent Low Wolf film franchise, where he competes with his peers in a wild competition. One of the better characters to come across since the film is to solve many clues and discover the identity of Low Wolf (a teenage slave assassin).



Night

June 2000 Volume 10 Number 1

—Thought you could get your hands on *Bliss Cops*, the game has been delayed, yet again.

Due originally for release almost a year ago, the game was among the countless when the fifth finger of fate struck. Arizona Software Limited, the programmer, who had worked to getting the game two weeks before being ready, have sided out on the company. This means a new programmer will have to finish the project and thus we shall see our Micro Cops until the beginning of September.

Commodore Where Are You?

As we continually get asked for the telephone numbers of Commanders U. R., it is best if we just post a hint: Tel. 04220 270000

The Siblings: *By Alan Brinkley*



Emlyn Hughes



INTERNATIONAL



**GAME
OF THE YEAR!**

**WINNER OF CCI OSKARS FOR
BEST SPORTS SIMULATION
AND BEST GAME OF THE YEAR!**

Quite simply the best football
simulation available for the 64
Nothing short of superb

ZZAP 64

**AVAILABLE NOW FOR
COMMODORE 64
£9.95 tape £12.95 disk**
Coming soon for Spectrum & Amstrad

Audiogenic

Audiogenic Software Limited, Winchester House,
Canning Road, Harrow HA3 7SL, England

Order by phone on 01 861 1166

